Journal of
Marine Science and Engineering

MDPI

Article

Optimization of Integrated Tugboat–Berth–Quay Crane Scheduling in Container Ports Considering Uncertainty in Vessel Arrival Times and Berthing Preferences

Liangyong Chu [1,2,*], Jiawen Zhang [1], Xiuqian Chen [1] and Qing Yu [1,*]

[1] School of Navigation, Jimei University, Xiamen 361021, China; zjw1561577580@163.com (J.Z.); 18350081635@163.com (X.C.)
[2] Fujian Shipping Research Institute, Xiamen 361021, China
* Correspondence: chuliangyong@163.com (L.C.); qing.yu@jmu.edu.cn (Q.Y.)

Abstract: Influenced by the dynamics of supply and demand, the demand for maritime transport has been increasing annually, putting significant pressure on container ports. To alleviate this pressure, a new mixed-integer programming model for the integrated scheduling of tugboats, berths, and quay cranes has been established. This model considers the uncertainties in vessel arrival times, vessel berthing preferences, time-varying quay crane availability, and the constraint that quay cranes cannot cross each other. The objective is to minimize the total costs including fuel consumption during port stays, delays and waiting times for berthing and departure, berthing deviation costs, tugboat assistance costs, and quay crane handling costs. To obtain high-quality solutions, an adaptive large neighborhood search (ALNS) algorithm was employed to solve the model. The algorithm incorporated five destruction operators and five repair operators that were specifically designed to enhance the solution accuracy and efficiency for the integrated scheduling problem. Several case studies of varying scales, based on a port in China, were used to validate the effectiveness of the proposed model and algorithm. The experimental results demonstrate the model's validity and show that the ALNS algorithm designed for the integrated scheduling problem outperformed CPLEX and other algorithms in terms of the accuracy and efficiency. Finally, a sensitivity analysis of the key parameters provides recommendations for the integrated scheduling of tugboats, berths, and quay cranes, offering valuable insights for port operations.

Keywords: waterway transportation; integrated tugboat–berth–quay crane scheduling; mixed-integer linear programming model; adaptive large neighborhood search algorithm; uncertainty in vessel arrival times; berthing preferences

check for updates

Citation: Chu, L.; Zhang, J.; Chen, X.; Yu, Q. Optimization of Integrated Tugboat–Berth–Quay Crane Scheduling in Container Ports Considering Uncertainty in Vessel Arrival Times and Berthing Preferences. *J. Mar. Sci. Eng.* **2024**, *12*, 1541. https://doi.org/10.3390/jmse12091541

Academic Editor: Claudio Ferrari

Received: 6 August 2024
Revised: 23 August 2024
Accepted: 25 August 2024
Published: 4 September 2024

1. Introduction

With the rapid development of the global economy and the continuous increase in international trade volume, ports, as crucial nodes in the logistics and transportation system, play an essential role in the efficiency and service quality. In port operations, the scheduling of berths, quay cranes, and tugboats is a critical factor affecting port operational efficiency. Therefore, optimizing the scheduling of these resources to improve port throughput capacity and service levels has become a focal point of interest for both academics and practitioners.

Berth allocation mainly involves the mooring arrangement of vessels at the port. An optimized berth allocation can effectively reduce the vessel waiting times and improve the overall utilization rate of the port. Quay crane scheduling relates to the loading and unloading efficiency of cargo between the vessel and the dock, which is essential to ensure the quick turnaround of port operations. Tugboat scheduling affects not only the speed of vessel berthing and departure, but also the safety of port operations. In-depth research on the scheduling of port resources has been conducted by some scholars, but most of these studies are limited to the optimization of a single resource, lacking systematic research on the integrated scheduling of berths, quay cranes, and tugboats. Single-resource optimization cannot meet the needs of modern port efficient operations; integrated scheduling is a key approach to enhancing the overall port efficiency. Therefore, research on the integrated scheduling of berths, quay cranes, and tugboats has significant theoretical and practical implications in addressing bottlenecks in actual port operations. This work seeks to fill this research gap by constructing a comprehensive integrated scheduling model and using optimization algorithms to improve the collaborative utilization efficiency of port resources and provide theoretical support and practical guidance for port operations, facilitating the development toward intelligent and refined port management.

To improve the efficiency of port quayside operations, a mixed-integer programming model for the integrated scheduling of tugboats, berths, and quay cranes is proposed in this paper. The objective was to minimize the total costs including fuel consumption costs during the vessel's stay in port, penalties for waiting and delayed departure, berth deviation penalties, tugboat fuel costs, and quay crane loading and unloading costs. Subsequently, the integrated scheduling model was linearized, converting it into a linear mixed-integer programming model. The CPLEX solver and the adaptive large neighborhood search (ALNS) algorithm were then employed to solve it. The effectiveness of the model and method was verified in the experimental results section. The contributions of this paper are as follows:

(1) A linear integer programming model for the integrated scheduling of tugboats, berths, and quay cranes was constructed, which can yield the optimal berth allocation scheme using CPLEX for small-scale instances.
(2) An adaptive large neighborhood search algorithm tailored to the characteristics of the integrated scheduling problem of tugboats, berths, and quay cranes was developed. For large-scale instances, the adaptive large neighborhood search algorithm can obtain near-optimal solutions in a relatively short time.
(3) Practical constraints such as the uncertainty of vessel arrival times, vessel berthing preferences, the non-overlapping of quay cranes, and partially time-varying constraints of quay cranes were considered, thereby enhancing the applicability of the integrated scheduling scheme.
(4) A continuous berth allocation layout was implemented, improving the utilization of the port quay space.
(5) The model's validity and the algorithm's efficiency were verified through data experiments.

The rest of this paper is organized as follows. Section 2 reviews the related literature on port berth, quay crane, and tugboat scheduling, summarizing the deficiencies of the existing research. Section 3 constructs the linear integer programming model for the integrated scheduling of tugboats, berths, and quay cranes. Section 4 designs the solution method for the model. Section 5 presents a case analysis to verify the effectiveness of the model and algorithm. Section 6 summarizes the paper.

2. Literature Review

The literature related to the scheduling of tugboats, berths, and quay cranes is summarized in this section. It provides a summary of the relevant studies in four aspects: berth allocation, integrated scheduling of berths and quay cranes, tugboat scheduling, and integrated scheduling of berths, tugboats, and quay cranes.

2.1. Research on Berth Allocation

Berth allocation can be categorized into three types: discrete, hybrid, and continuous. In the research on discrete berth allocation problems, Imai et al. [1] aimed to minimize the waiting and handling times for all vessels by constructing a nonlinear integer programming model and designing a Lagrangian relaxation-based heuristic algorithm to solve it. Prencipe et al. [2] focused on minimizing the in-port time of vessels by developing a linear mixed-integer programming model and designing a swarm optimization algorithm to solve it. Iris et al. [3] developed a set-packing model for the strategic berth allocation problem at large container ports and demonstrated the model's excellence through the analysis of numerous examples of varying scales.

In hybrid berth allocation, a large vessel can occupy multiple berths simultaneously if there is sufficient space, and several small vessels can also dock at one berth concurrently. Umang et al. [4] constructed a mixed-integer linear programming model with the objective of minimizing the in-port time of all vessels and designed a squeaky wheel optimization algorithm for its solution. Kordić et al. [5] aimed to minimize the total in-port costs of all vessels by developing a linear mixed-integer programming model and designing an exact algorithm based on the precipitation method to solve it. Venturini et al. [6] were the first to propose the concept of the multi-port berth allocation problem, considering both vessel speed optimization and emissions. They developed a MILP model for this problem and demonstrated the model's effectiveness and applicability through extensive data experiments.

For continuous berth allocation problems, Ernst et al. [7] studied berth allocation in export ports and established a two-stage linear mixed-integer programming model, introducing two efficient inequalities to enhance the solvability of the model. Cheimanoff et al. [8] also researched berth allocation in export ports, creating a mixed-integer linear programming model, designing a variable neighborhood search algorithm to solve it, and employing machine learning methods to adjust the algorithm parameters. Martin-Iradi et al. [9] investigated continuous berth allocation problems in multi-port scenarios, developing a mixed-integer linear programming model for this problem.

2.2. Integrated Scheduling of Berths and Quay Cranes

The integrated scheduling of berths and quay cranes also requires the consideration of the division of the quay line, with related research primarily focusing on discrete and continuous types. Xiang et al. [10] studied the discrete berth and quay crane integrated scheduling problem, addressing uncertainties through the reservation of time slots. Wang et al. [11] researched the continuous berth and quay crane integrated scheduling problem under various uncertainties. In the continuous division, the quay line is divided into multiple smaller berths, and vessels need to occupy several berths when docking, thereby improving the utilization of the quay line.

Research on the integrated scheduling of berths and quay cranes involves numerous variables and constraints, making it challenging to solve all variables simultaneously. Therefore, related research often adopts a phased approach. Ji et al. [12] solved the berth and quay crane integrated scheduling problem in two stages: the first stage determines the berthing position, berthing time, and the number of quay cranes assigned to the vessel, while the second stage allocates specific quay cranes to the corresponding vessels. Liu et al. [13] used a rolling horizon strategy to solve the integrated scheduling problem of sea-rail intermodal ports in two stages: the first stage determines the berth occupied by the vessel and the number of quay cranes assigned to each vessel, and the second stage uses a heuristic algorithm to solve the time variables. Chargui et al. [14] employed a decomposition algorithm to divide the problem into a master problem and multiple scenario sub-problems: the first stage solves the deterministic master problem, and the second stage decides whether to solve the sub-problems based on the results of the master problem.

The research on the integrated scheduling of berths and quay cranes must consider time-invariant or time-variant constraints on the number of quay cranes. The time-invariant constraint means that the number of quay cranes assigned to the vessel remains unchanged during its berthed period; the time-variant constraint means that the number of quay cranes assigned to the vessel can change during its berthed period. Yu et al. [15] studied the integrated scheduling of berths and quay cranes under time-invariant constraints, aiming to balance the economic and environmental benefits. Li et al. [16] studied the integrated berth and quay crane scheduling problem by considering quay crane maintenance issues and employed a time-varying strategy for the number of quay cranes. Iris et al. [17] compared the integrated scheduling of berths and quay cranes under time-variant and time-invariant strategies and found that the scheduling efficiency of time-variant strategies was higher than that of time-invariant strategies due to the greater flexibility and efficiency of fully time-variant strategies, which could improve the utilization of quay cranes. When considering time-invariant constraints, the scheduling efficiency is generally lower than when considering time-variant constraints. Time-variant strategies can be fully time-variant or partially time-variant. The modeling and solving process for fully time-variant strategies is more complex, while the complexity of partially time-variant strategies is not significantly higher than that of time-invariant strategies.

When vessels enter and leave port channels, it is necessary to consider whether the channel and berth depths meet the vessel's draft requirements. Due to tidal factors, the depths of the port berths and channels change periodically, significantly affecting vessel entry, departure, and berthing. Malekahmadi et al. [18] studied the impact of tidal factors on the berth depth, emphasizing that if the berth depth does not meet the vessel's draft requirement, the vessel cannot dock. Krimi et al. [19] studied the integrated scheduling of berths and quay cranes in export ports, where vessels are unaffected by tides when entering the port, but must take advantage of the tides when departing.

Uncertainty factors are inevitable in port operations, and research on the integrated scheduling of berths and quay cranes should consider these uncertainties. Iris and Lam et al. [20] addressed planned uncertainties by setting buffer times and using different buffer times to handle different levels of uncertainty. Liu et al. [13] incorporated uncertainty sets into the model to address uncertainties related to vessels and quay cranes and introduced control parameters to manage the degree of uncertainty in the model.

The integrated scheduling problem of berths and quay cranes is an NP-hard problem. The type of model determines the solution method. For a mixed-integer linear programming (MILP) model, heuristic algorithms and CPLEX can be used; for a nonlinear mixed-integer programming model, only heuristic algorithms can solve the model. He et al. [21] considered the fully time-variant strategy for the number of quay cranes, establishing a mixed-integer nonlinear programming model and solving it with a memetic algorithm. Wang et al. [22] established a MILP model considering both single-rate and piece-rate port carbon tax policies, employing equivalent and relaxed constraints to simplify the model, and solving it with CPLEX. Cheimanoff et al. [23] developed a MILP model for the integrated scheduling of berths and quay cranes, employing CPLEX and a variable neighborhood search algorithm to solve the model. Iris et al. [24] developed a MILP model for the integrated berth and quay crane scheduling problem. They improved the model's solving speed by introducing valid inequalities and designed an adaptive large neighborhood search (ALNS) algorithm based on the problem characteristics to solve the model. Finally, data experiments demonstrated the efficiency of the model improvements and the effectiveness of the algorithm designed in their study.

2.3. Tugboat Scheduling

The tugboat scheduling problem can be divided into static and dynamic tugboat scheduling problems. The static tugboat scheduling problem assumes that all vessels have already arrived at the port before the scheduling begins. Xu et al. [25] studied the static tugboat scheduling problem by considering the vessel relocation, establishing a three-stage model including vessel berthing, shifting, and departure, and designed a hybrid simulated annealing-ant colony optimization algorithm to solve the model.

The dynamic tugboat scheduling problem allows vessels to arrive at the port at any time during the planning period, with all vessel arrival times known. Wei et al. [26] pointed out that static scheduling does not conform to practical scenarios and proposed a mixed-integer programming model where vessels can arrive at different times during the planning period, designing a customized heuristic algorithm to solve the model. Sun et al. [27] studied the dynamic tugboat scheduling problem in large-scale ports by considering the tugboat operation uncertainties and tugboat transit area restrictions. A mixed-integer programming model was developed, and a reverse operation-based genetic algorithm was designed to solve the model. The algorithm's effectiveness was validated with data from Zhoushan Port. Li et al. [28] considered the uncertainty of vessel arrival times, established a stability model with the objective of minimizing the sum of vessel waiting costs and tugboat fuel consumption, and then solved the model using a reinforcement learning algorithm. Zhong et al. [29] considered the impact of tidal factors on tugboat assistance for vessel entry, created a multi-objective mixed-integer linear programming model aiming to minimize tugboat handling time and fuel consumption, and designed an NSGA-II algorithm to solve the model. The algorithm and model's effectiveness were validated through experiments using data from Guangzhou Port. Kang et al. [30] studied the tugboat scheduling problem considering uncertainties in vessel arrival times and tugboat towing processes, established a mixed-integer linear programming model for tugboat scheduling, and designed a self-organizing map (SOM) algorithm to solve the model.

Wei et al. [31] noted the path point selection patterns in port scheduling practices and developed a mixed-integer linear programming model incorporating these patterns, designing a branch-and-bound algorithm with six different effective inequalities to solve the model. The algorithm and model's effectiveness were validated through experiments with data from Singapore Port. Jia et al. [32] considered the vessel berthing plans and different vessel entry paths with the objectives of minimizing vessel delays, tugboat operation costs, and unserved vessels. A mixed-integer linear programming model was established, and a heuristic method based on Lagrangian relaxation and Benders decomposition was designed to solve the model. The heuristic's effectiveness was validated with data from Shanghai. Wang et al. [33] aimed to improve tugboat utilization by developing a mixed-integer programming model for tugboat scheduling by considering multiple vessel entry path modes. An adaptive large neighborhood search algorithm incorporating a feasible path detection procedure was designed to solve the model, and the model's effectiveness was validated through case studies.

2.4. Integrated Scheduling of Tugboats, Berths, and Quay Cranes

There is limited research on the integrated scheduling of tugboats, berths, and quay cranes. To the best of our knowledge, Yang et al. [34] are the only researchers who have studied this problem. The integrated scheduling of tugboats, berths, and quay cranes, considering the economic benefits, was investigated. A nonlinear mixed-integer programming model for the integrated scheduling problem was developed, and an adaptive satin bower bird optimizer algorithm integrating chaotic mapping and quantum computing strategies was designed to solve the model. In the tugboat scheduling part of their study, each berthing and unberthing operation required a certain number of tugboats with specific horsepower to be allocated to the vessel. The time required for berthing and unberthing varied depending on the number of tugboats assigned. For berth allocation, the study focused on discrete berth allocation layouts. In quay crane scheduling, a time-invariant strategy for the number of quay cranes was employed, ensuring that the quay cranes did not cross each other. Our study also investigated the integrated scheduling of tugboats, berths, and quay cranes. Table 1 compares our study with the study by Yang et al. The key differences include accounting for the uncertainty of vessel arrival times, utilizing a continuous berth layout, implementing a time-variant strategy for quay crane allocation, developing a linear mixed-integer programming model, and employing an adaptive large neighborhood search algorithm to solve the model.

Table 1. Comparison of the study details.

Reference	Conditions	Berth	Quay Crane	Model	Method
Yang et al. (2024) [34]	Vessel berthing preference	Discrete	Quay cranes cannot cross each other; quay crane quantity is time-invariant	Nonlinear mixed-integer programming model	Adaptive satin bower bird optimizer algorithm
This paper	Vessel berthing preference; vessel arrival time uncertainty	Continuous	Quay cranes cannot cross each other; Quay crane quantity is time-variant	Linear mixed-integer programming model	CPLEX, CPLEX, adaptive large neighborhood search algorithm

3. Model

3.1. Problem Description

As shown in Figure 1, the integrated scheduling problem of tugboats, berths, and quay cranes studied in this paper can be described as follows. After arriving at the port, vessels enter the anchorage area. If the number of available tugboats meets the required number for assisting a ship in entering the port, the corresponding tugboats will be allocated to facilitate its entry. If the available tugboats cannot meet the number requirements, the vessel waits at the anchorage until sufficient tugboats are available. Similarly, if there are not enough continuous berths available on the quay line, the vessel also needs to wait at the anchorage until sufficient berths are free to enter the port. Once berthed, quay cranes are immediately allocated to the vessel for loading and unloading operations. If the vessel's berthing position deviates from its preferred berth, additional container truck transportation costs are incurred, referred to as berth deviation penalties in this paper. Since all quay cranes move on the same track, they cannot cross each other. If the number of available tugboats meets the required number for assisting a ship in leaving the port, the corresponding tugboats will be allocated to facilitate its departure. If the available tugboats do not meet the number requirements, the vessel waits at the berth until enough tugboats are available for departure. Time-variant constraints on the number of quay cranes were considered in this work. During the vessel's waiting period after the completion of loading and unloading operations, the quay cranes allocated to this vessel can move to other berths to handle other newly berthed vessels. This study primarily addresses the uncertainty in vessel arrival times by implementing a buffer period. After a vessel completes its loading and unloading operations, the berth it occupied must remain vacant for a designated buffer time before accommodating a new incoming vessel. As illustrated in Figure 1, after entering the port, Ship 3 can only dock in the green-marked mooring area near the shoreline and is prohibited from docking in the yellow-marked non-mooring area because Ship 5 has just vacated that zone, which must remain unavailable for other vessels until the buffer time has expired. This buffer period allows for adjustments in the loading and unloading schedules when there are fluctuations in vessel arrival times. The larger the buffer interval, the higher the stability of the model.

3.2. Model Assumptions

(1) The port water depth meets the draft requirements of all vessels; (2) vessels will not be relocated, and each vessel can berth only once; (3) the berthing of vessels must meet physical conditions; (4) each vessel has a preferred berth; (5) the safe distance between vessels is included in the vessel length; (6) the number of quay cranes allocated to each vessel has both upper and lower limits; (7) vessel deviation from the preferred berth will incur additional container truck scheduling costs; (8) vessels start loading and unloading services immediately after berthing; (9) the number of tugboats allocated to a ship should meet the required number necessary for the ship's entry or departure from the port; and (10) each tugboat provides the same amount of horsepower.

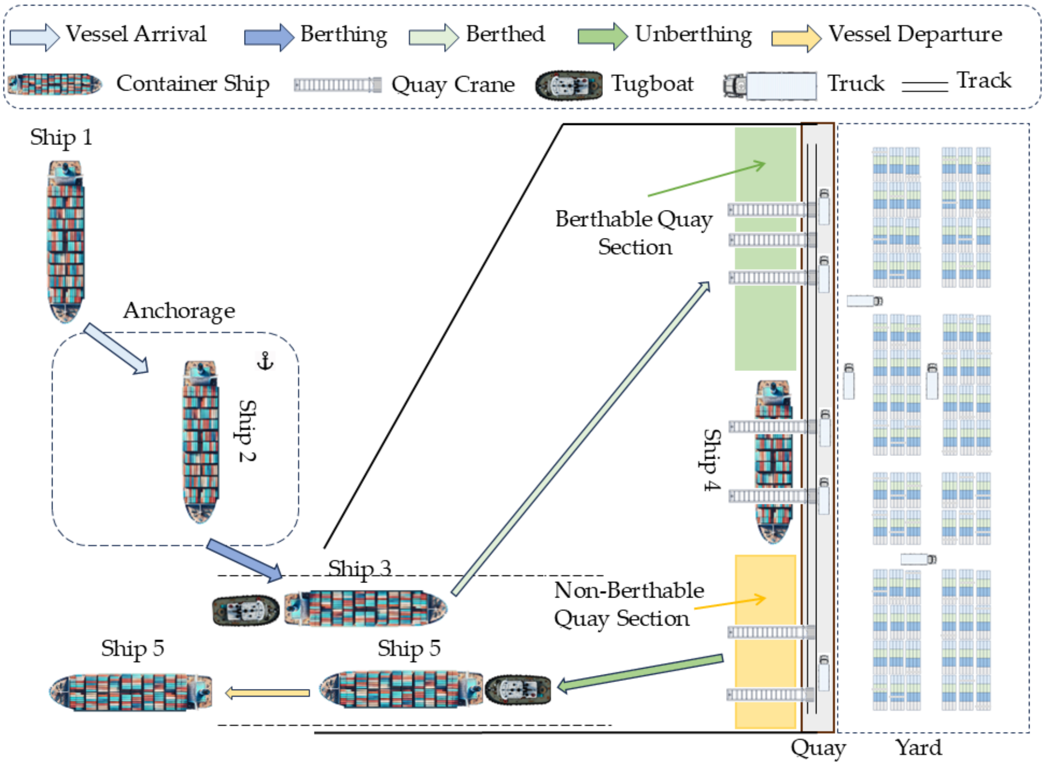


Figure 1. Schematic of integrated scheduling for tugboats, berths, and quay cranes.

3.3. Model Parameters

The parameters and variables of the integrated scheduling model for tugboats, berths, and quay cranes constructed in this paper are shown in Table 2.

Table 2. Model parameters and variables.

Notation	Sets
v, v'	Vessel index
t	Time period index
q	Quay index
s	Tugboat index
V	Set of incoming vessels, $V = \{1,2, \ldots \mid V \mid\}$
B	Set of berths on the quay, $B = \{1,2, \ldots \mid B \mid\}$
Q	Set of quay cranes, $Q = \{1,2, \ldots \mid Q \mid\}$
T	Set of time periods, $T = \{1, 2, \ldots \mid T \mid\}$
S	Set of tugboats, $S = \{1, 2, \ldots \mid S \mid\}$
C_1	Parameter: vessel in-port fuel consumption rate
C_2	Parameter: vessel delay departure and waiting cost per time unit
C_3	Parameter: penalty cost per unit deviation from preferred berth
C_4	Parameter: tugboat fuel consumption rate
C_5	Parameter: quay crane usage rate
M	Parameter: a sufficiently large number
GAP	Parameter: the parameter represents the minimum buffer time interval between successive vessel berths in the same berth space.
SN_v	Parameter: minimum number of tugboats required for vessel entry and exit
AT_v	Parameter: arrival time of vessel v
L_v	Parameter: length of vessel v
PB_v	Parameter: preferred berth of vessel v
LDT_v	Parameter: latest departure time of vessel v
QU_v	Parameter: maximum number of quay cranes that can be allocated to vessel v
QL_v	Parameter: minimum number of quay cranes that can be allocated to vessel v

Table 2. *Cont.*

Notation	Sets
FT_v	Parameter: quay crane service hours required for vessel v
$S1_v$	Variable: time when vessel v starts berthing from the anchorage
ET_{vs}	Variable: time required for vessel s to enter the port when v tugboats are allocated
$LTvs$	Variable: time required for vessel v to exit the port when s tugboats are allocated
$S2_v$	Variable: time when vessel v starts unberthing from the berth
BT_v	Variable: berthing time of vessel v
DT_v	Variable: departure time of vessel v
HT_v	Variable: loading and unloading time of vessel v at the quay
LQ_v	Variable: index of the leftmost quay crane allocated to vessel v
RQ_v	Variable: index of the rightmost quay crane allocated to vessel v
EHT	Variable: time when vessel v finishes loading and unloading
P_v	Variable: berthing position of vessel v
DS_v	Variable: deviation distance of vessel v from the preferred berth
R_v	Variable: delayed departure time of vessel v
$X_{vv'}$	Variable: $X_{vv'}$ = 1 if vessel v' berths later than vessel v depart, otherwise $X_{vv'}$ = 0
$Y_{vv'}$	Variable: $Y_{vv'}$ = 1 if vessel v is completely berthed to the left of vessel v', otherwise $Y_{vv'}$ = 0
$Z_{vv'}$	Variable: $Z_{vv'}$ = 1 if vessel v' berths later than vessel v finishes loading and unloading, otherwise $Z_{vv'}$ = 1
Q_{vq}	Variable: AQ_{vq} = 1 if q quay cranes are allocated to vessel v, otherwise AQ_{vq} = 1
SA_{vs}	Variable: SA_{vs} = 1 if s tugboats are allocated to incoming vessel v, otherwise SA_{vs} = 0
SE_{vs}	Variable: SE_{vs} = 1 if s tugboats are allocated to outgoing vessel v, otherwise SE_{vs} = 0
RT_{vt}	Variable: RT_{vt} if tugboats are allocated to incoming vessel v in period t, otherwise RT_{vt} = 0
CT_{vt}	Variable: CT_{vt} = 1 if tugboats are allocated to outgoing vessel v in period t, otherwise CT_{vt} = 0
$AU1_{vq}$	Variable: auxiliary variable introduced to linearize the model
$AU2_{vst}$	Variable: auxiliary variable introduced to linearize the model
$AU3_{vst}$	Variable: auxiliary variable introduced to linearize the model

3.4. Scheduling Model

3.4.1. Objective Function

$$F = F_1 + F_2 + F_3 + F_4 + F_5 \tag{1}$$

$$F_1 = C_1 \sum_{\mathrm{v} \in V} (DT_v - AT_v) \tag{2}$$

$$F_2 = C_2 \sum_{v \in V} [R_v + S1_v - AT_v + S2_v - EHT_v] \tag{3}$$

$$F_3 = C_3 \sum_{v \in V} DS_v \tag{4}$$

$$F_4 = C_4 \sum_{v \in V} \sum_{s \in S} (sET_{vs}SA_{vs} + sLT_{vs}SE_{vs}) \tag{5}$$

$$F_5 = C_5 \sum_{v \in V} \sum_{q \in Q} \left(qAQ_{vq}HT_v\right) \tag{6}$$

In the objective function, F_1 represents the fuel consumption costs of all vessels during their stay in the port, F_2 represents the total penalties for the delayed departure and waiting time of vessels, F_3 represents the total penalties for the berth deviation of vessels, F_4 represents the costs incurred by tugboats assisting vessels in entering and leaving the port, and F_5 represents the costs incurred by quay cranes loading and unloading vessels

3.4.2. Tugboat-Related Constraints

$$S1_v \geq AT_v, \forall v \in V \tag{7}$$

$$BT_v = S1_v + \sum_{s \in S} SA_{vs} ET_{vs}, v \in V \tag{8}$$

$$\sum_{s \in S} SA_{vs} = 1, \forall v \in V \tag{9}$$

$$tRT_{vt} + M(1 - RT_{vt}) \geq S1_v, v \in V, t \in T \tag{10}$$

$$\sum_{t \in T} RT_{vt} = BT_v - S1_v, \forall v \in V \tag{11}$$

$$(t+1)RT_{vt} \leq BT_v, \forall v \in V, t \in T \tag{12}$$

$$\sum_{s \in S} sSA_{vs} \geq SN_v, \forall v \in V \tag{13}$$

$$S2_v \geq EHT_v, \forall v \in V \tag{14}$$

$$DT_v = S2_v + \sum_{s \in S} SE_{vs} LT_{vs}, v \in V \tag{15}$$

$$\sum_{s \in S} SE_{vs} = 1, \forall v \in V \tag{16}$$

$$tCT_{vt} + M(1 - CT_{vt}) \geq S2_v, v \in V, t \in T \tag{17}$$

$$\sum_{t \in T} CT_{vt} = DT_v - S2_v, \forall v \in V \tag{18}$$

$$(t+1)CT_{vt} \leq DT_v, \forall v \in V, t \in T \tag{19}$$

$$\sum_{s \in S} sSE_{vs} \geq SN_v, v \in V \tag{20}$$

$$\sum_{v \in V} \sum_{s \in S} sSA_{vs} RT_{vt} + \sum_{v \in V} \sum_{s \in S} sSE_{vs} CT_{vt} \leq |S|, \forall t \in T \tag{21}$$

Constraint (7) ensures that vessels can only enter the port and berth after arriving. Constraint (8) ensures that the berthing time with tugboat assistance is related to the number of tugboats assigned to the vessel. Constraint (9) ensures that the number of tugboats assigned to a vessel remains constant during the tugboat-assisted berthing process. Constraints (10) to (12) ensure the continuity of the vessel's entry and berthing times. Constraint (13) ensures that the number of tugboats allocated to incoming ships satisfies the tugboat requirements for these ships. Constraint (14) ensures that vessels can only depart after completing loading and unloading operations. Constraint (15) ensures that the unberthing time with tugboat assistance is related to the number of tugboats assigned to the vessel. Constraint (16) ensures that the number of tugboats assigned to a vessel remains constant during the tugboat-assisted unberthing process. Constraints (17) to (19) ensure the continuity of the vessel's unberthing and departure times. Constraint (19) ensures that the number of tugboats allocated to departing ships satisfies the tugboat requirements for these ships. Constraint (20) ensures that the number of tugboats assisting vessels in entering and leaving the port at any given time does not exceed the maximum number of tugboats available at the port.

3.4.3. Vessel-Related Constraints

$$S2_v \geq EHT_v, \forall v \in V \tag{22}$$

$$EHT_v >= BT_v + HT_v, v \in V \tag{23}$$

$$DS_v \geq P_v - PB_v, v \in V \tag{24}$$

$$DS_v \geq PB_v - P_v, v \in V \tag{25}$$

$$S2_{\text{v}} + GAP \leq BT_{v'} + (1 - X_{vv'}), \forall v, v' \in V, v \neq v' \tag{26}$$

$$P_v + L_v \leq P_{v'} + (1 - Y_{vv'}), \forall v, v' \in V, v \neq v' \tag{27}$$

$$X_{vv'} + X_{v'v} + Y_{vv'} + Y_{v'v} \geq 1, \forall v, v' \in V, v \neq v' \tag{28}$$

$$P_v + L_v \leq |B|, v \in V \tag{29}$$

$$R_v \geq 0, v \in V \tag{30}$$

$$R_v \geq DT_v - LDT_v, v \in V \tag{31}$$

$$DT_v \leq H - 1 \tag{32}$$

Constraint (22) ensures that vessels can only depart after completing loading and unloading operations. Constraint (23) ensures that vessels receive loading and unloading services immediately after berthing and defines the relationship between the vessel's berthing time, loading and unloading time, and the completion time of these operations. Constraints (24) to (25) ensure that the berth deviation distance for vessels is positive. Constraints (27) to (28) ensure that the quay space and time occupied by any two berthed vessels do not overlap. Constraint (29) ensures that vessels can only berth within the quay area. Constraints (30) to (31) ensure that the delayed departure time for vessels that did not experience a delayed departure is zero, and that it is positive for vessels that did experience a delayed departure. Constraint (32) ensures that the vessel's departure time is within the planning period.

3.4.4. Quay Crane-Related Constraints

$$\sum_{q \in Q} qAQ_{vq}HT_v \geq FT_v, \forall v \in V \tag{33}$$

$$\sum_{q \in Q} AQ_{vq} = 1, \forall v \in V \tag{34}$$

$$LQ_{\text{v}} \leq RQ_v, \forall v \in V \tag{35}$$

$$\sum_{q \in Q} qAQ_{vq} = RQ_v - LQ_v + 1, v \in V \tag{36}$$

$$\sum_{q \in Q} qAQ_{vq} \leq QU_v, v \in V \tag{37}$$

$$\sum_{q \in Q} qAQ_{vq} \geq QL_v, v \in V \tag{38}$$

$$RQ_v \leq |Q| - 1, v \in V \tag{39}$$

$$EHT_{\text{v}} \leq BT_{v'} + M(1 - Z_{vv'}), \forall v, v' \in V, v \neq v' \tag{40}$$

$$RQ_v \leq LQ_{v'} + M * (1 - Y_{vv'}) + M * Z_{vv'} + M * Z_{v'v} - 1, \forall v, v \in V, v \neq v' \tag{41}$$

Constraint (33) ensures that vessels can complete loading and unloading tasks within the given time and conditions. Constraint (34) ensures that the number of quay cranes assigned to a vessel is unique. Constraints (35) to (36) ensure that the quay crane numbers assigned to the same vessel are consecutive. Constraints (37) to (38) ensure that the number of quay cranes assigned to a vessel is within the available range of quay cranes for that vessel. Constraint (39) ensures that the total number of quay cranes assigned to all vessels does not exceed the total number of quay cranes. Constraints (40) to (41) ensure that the quay cranes assigned to a vessel can transfer along the track to handle newly berthed vessels after completing the loading and unloading operations for the current vessel, even if the vessel needs to wait at the berth for sufficient tugboats to become available for departure. This ensures that the number of quay cranes assigned to a vessel meets the partial time-variant constraints.

$$X_{vv'}, Y_{vv'}, Z_{vv'} = \{0, 1\}, v, v \in V, v \neq v' \tag{42}$$

$$AQ_{vq} = \{0, 1\}, \forall v \in V, q \in Q \tag{43}$$

$$AT1_{vs}, AT2_{vs} = \{0, 1\}, v \in V, s \in S \tag{44}$$

$$AT1_{vt}, AT2_{vt} = \{0, 1\}, v \in V, t \in T \tag{45}$$

$$\begin{aligned} &S1_v, S2_v, BT_v, DT_v, HT_v, LQ_v, RQ_v, EHT_v, P_v, STI_v, \\ &ETI_v, DS_v, R_v = \{0, 1, 2 \ldots, \infty\}, \forall v \in V \end{aligned} \tag{46}$$

Constraints (42) to (46) ensure the type and range of values for the decision variables.

3.5. Model Linearization

In Equations (6) and (42), $AQ_{vq}HT_v$ is nonlinear. To linearize it, we need to introduce an auxiliary variable $AU1_{vq}$ and use $AU1_{vq}$ to replace $AQ_{vq}HT_v$. To ensure equivalence, additional constraints (47) to (51) need to be introduced.

$$AU1_{vq} \geq HT_v - M\big(1 - AQ_{vq}\big), \forall v \in V, q \in Q \tag{47}$$

$$AU1_{vq} \leq HT_v, \forall v \in V, q \in Q \tag{48}$$

$$AU1_{vq} \leq MAQ_{vq}, \forall v \in V, q \in Q \tag{49}$$

$$AU1_{vq} \geq -MAQ_{vq}, \forall v \in V, q \in Q \tag{50}$$

$$AU1_{vq} = \{0, 1, 2 \ldots \infty\}, \forall v \in V, q \in Q \tag{51}$$

In Equation (21), $SA_{vs}RT_{vt}$ and $SE_{vs}CT_{vt}$ need to be linearized by introducing auxiliary variables $AU2_{vst}$ and $AU3_{vst}$ to replace $SA_{vs}RT_{vt}$ and $SE_{vs}CT_{vt}$, respectively. To ensure equivalence, additional constraints (52) to (58) need to be introduced.

$$AU2_{vst} \leq SA_{vs}, \forall v \in V, s \in S, t \in T \tag{52}$$

$$AU2_{vst} \leq RT_{vt}, \forall v \in V, s \in S, t \in T \tag{53}$$

$$AU2_{vst} \geq SA_{vs} + RT_{vt} - 1, \forall v \in V, s \in S, t \in T \tag{54}$$

$$AU3_{vst} \leq SE_{vs}, \forall v \in V, s \in S, t \in T \tag{55}$$

$$AU3_{vst} \leq CT_{vt}, \forall v \in V, s \in S, t \in T \tag{56}$$

$$AU3_{vst} \geq SE_{vs} + CT_{vt} - 1, \forall v \in V, s \in S, t \in T \tag{57}$$

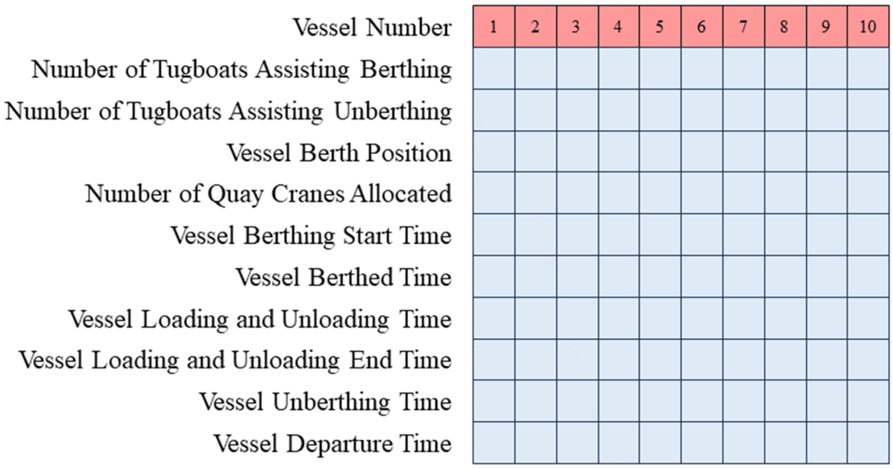
$$AU2_{vst}, AU3_{vst} = \{0, 1, 2 \ldots, \infty\}, \forall v \in V, s \in S, t \in T \tag{58}$$

4. Methodology

A mixed-integer linear programming (MILP) model for the integrated scheduling of tugboats, berths, and quay cranes was constructed in this paper. Therefore, the model can be solved using the CPLEX solver. However, due to the high complexity of the problem and the large number of variables, the solution time of CPLEX is significantly long, and the solution time increases exponentially with the size of the problem instance. This makes it challenging to apply to practical port scheduling. Hence, a heuristic algorithm is required to obtain a near-optimal integrated scheduling solution in a shorter time. To achieve high-quality integrated scheduling solutions within a shorter period, the adaptive large neighborhood search (ALNS) algorithm was employed to solve the model in this paper.

4.1. Solution Encoding

The solution encoding uses a multi-layer real number encoding format. As shown in Figure 2, the solution encoding for a set of 10 arriving vessels is illustrated.



Vessel Number	1	2	3	4	5	6	7	8	9	10
Number of Tugboats Assisting Berthing										
Number of Tugboats Assisting Unberthing										
Vessel Berth Position										
Number of Quay Cranes Allocated										
Vessel Berthing Start Time										
Vessel Berthed Time										
Vessel Loading and Unloading Time										
Vessel Loading and Unloading End Time										
Vessel Unberthing Time										
Vessel Departure Time										

Figure 2. Solution encoding format.

4.2. Initial Solution Generation

Based on information such as the vessels, quay cranes, and tugboats, an initial feasible solution was generated. The process for generating the initial solution is illustrated in Figure 3.

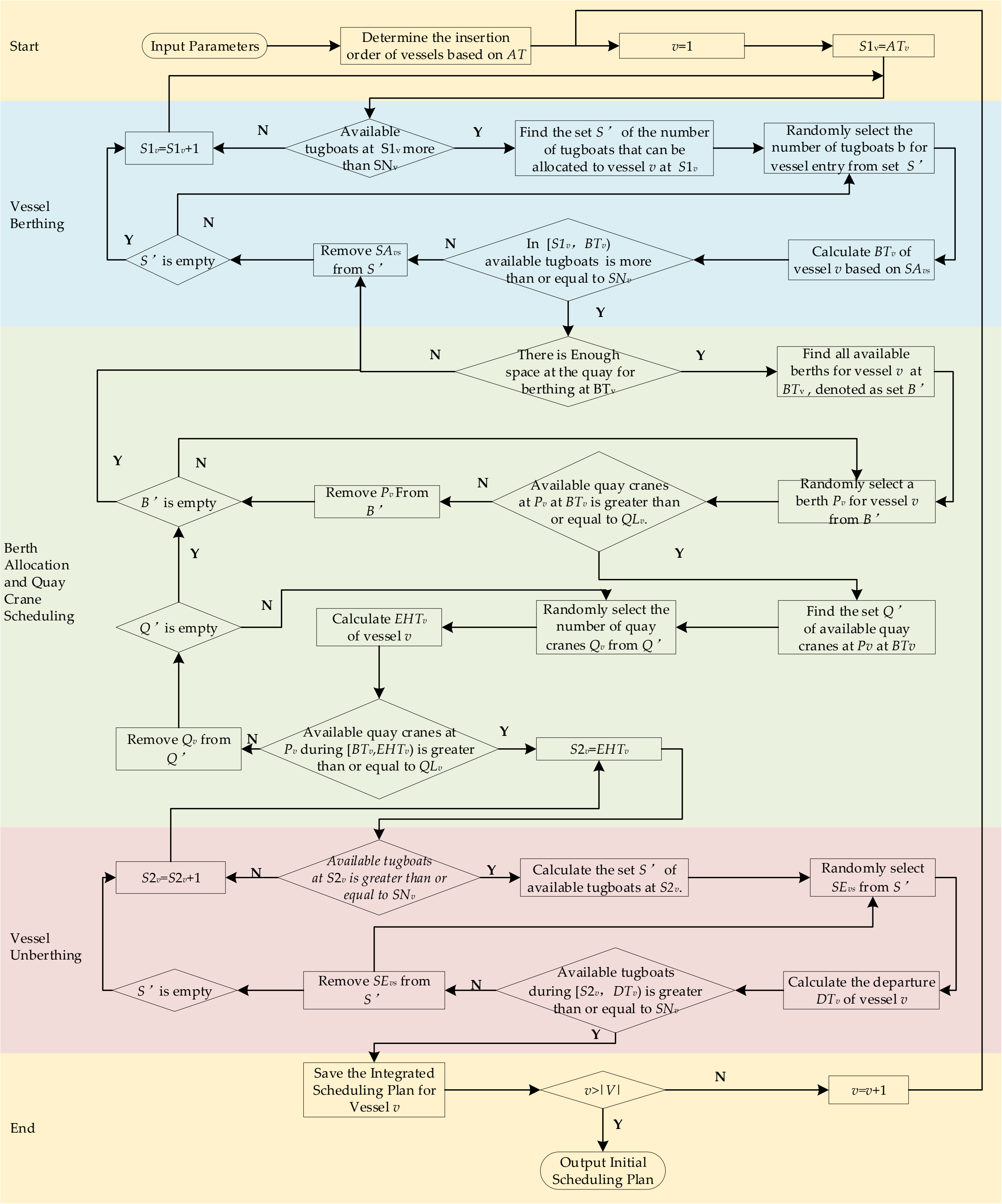

Start
Input Parameters
Determine the insertion order of vessels based on AT
v=1
S1v=ATv
Vessel Berthing
S1v=S1v+1
N
Available tugboats at S1v more than SNv
Y
Find the set S′ of the number of tugboats that can be allocated to vessel v at S1v
Randomly select the number of tugboats b for vessel entry from set S′
N
Y
S′ is empty
Remove SAvs from S′
N
In [S1v, BTv) available tugboats is more than or equal to SNv
Calculate BTv of vessel v based on SAvs
Y
Berth Allocation and Quay Crane Scheduling
N
There is Enough space at the quay for berthing at BTv
Y
Find all available berths for vessel v at BTv, denoted as set B′
Y
N
B′ is empty
Remove Pv From B′
N
Available quay cranes at Pv at BTv is greater than or equal to QLv.
Randomly select a berth Pv for vessel v from B′
Y
Y
N
Q′ is empty
Calculate EHTv of vessel v
Randomly select the number of quay cranes Qv from Q′
Find the set Q′ of available quay cranes at Pv at BTv
Remove Qv from Q′
N
Available quay cranes at Pv during [BTv,EHTv) is greater than or equal to QLv
Y
S2v=EHTv
Vessel Unberthing
S2v=S2v+1
N
Available tugboats at S2v is greater than or equal to SNv
Y
Calculate the set S′ of available tugboats at S2v.
Randomly select SEvs from S′
S′ is empty
Remove SEvs from S′
N
Available tugboats during [S2v, DTv) is greater than or equal to SNv
Calculate the departure DTv of vessel v
Y
End
Save the Integrated Scheduling Plan for Vessel v
v>|V|
N
v=v+1
Y
Output Initial Scheduling Plan

Figure 3. Initial solution generation process.

4.3. Destruction Operators

4.3.1. Random Destruction Operator

Randomly select σ vessels and remove their related information from the solution. The IDs of the removed vessels are recorded in ID_{del}, and the solution after removing σ vessels is denoted as X_{re}.

4.3.2. Temporal Destruction Operator

Calculate the temporal penalty F_2 for all vessels, and sequentially remove the vessels with the highest F_2 values until σ vessels have been removed. The IDs of the removed vessels are recorded in ID_{del}, and the solution after removing σ vessels is denoted as X_{re}.

4.3.3. Berth Deviation Destruction Operator

Calculate F_3 for all vessels, and sequentially remove the vessels with the highest F_3 values until σ vessels have been removed. The IDs of the removed vessels are recorded in ID_{del}, and the solution after removing σ vessels is denoted as X_{re}.

4.3.4. Quay Crane Margin Destruction Operator

Define the quay crane margin for vessel v as $ABQ_v = QU_v - \sum_{q\in Q}\left(qAQ_{vq}HT_v\right)$. Calculate the quay crane margin for all vessels, and sequentially remove pairs of vessels with the highest and lowest ABQ_v values until σ vessels have been removed. The IDs of the removed vessels are recorded in ID_{del}, and the solution after removing σ vessels is denoted as X_{re}.

4.3.5. Tugboat Margin Destruction Operator

Define the tugboat margin for vessel v as $ABT_v = 2 \mid S \mid - \sum_{s\in S}(sET_{vs} + sLT_{vs})$. Calculate the tugboat margin for all vessels, and sequentially remove pairs of vessels with the highest and lowest ABT_v values until σ vessels have been removed. The IDs of the removed vessels are recorded in ID_{del}, and the solution after removing σ vessels is denoted as X_{re}.

4.4. Repair Operators

4.4.1. Random Repair Operator

Randomly select a vessel from ID_{del} one by one and insert its schedule into X_{re} without conflicting with the schedules of the remaining vessels. The insertion process for a vessel is as follows: randomly allocate a certain number of tugboats to assist its entry into the port, randomly allocate a certain number of quay cranes to it, calculate the vessel's loading and unloading time in the port, then randomly select a berth for it, and randomly allocate a certain number of tugboats to assist its departure from the port. This process continues until all vessels in ID_{del} are inserted, completing the loading and unloading operations.

4.4.2. Berth Priority Repair Operator

Sort the vessels in ID_{del} by the degree of deviation from their preferred berths in descending order, and insert them into the solution in that order. The insertion process is as follows: randomly allocate a certain number of tugboats to assist the vessel's entry into the port, randomly allocate a certain number of quay cranes to the vessel, calculate the vessel's loading and unloading time in the port, and then assign the vessel to the berth closest to its preferred berth that can be processed the soonest. Ensure that there are no conflicts with the schedules of the remaining vessels during insertion, and randomly allocate a certain number of tugboats to assist with the vessel's departure from the port.

4.4.3. Time Priority Repair Operator

Sort the vessels in ID_{del} by their temporal penalty F_2 in descending order, and insert them into the solution in that order. The insertion process is as follows: determine the earliest possible entry time for the vessel, allocate the maximum available number of tugboats at that time to assist with the vessel's entry into the port, allocate the maximum available number of quay cranes at that time to the vessel, calculate the vessel's loading and unloading time in the port, and then assign the vessel to the berth that can be processed the soonest, ensuring no conflicts with other vessels. Allocate the maximum available number of tugboats at that time to assist with the vessel's departure from the port.

4.4.4. Quay Crane Margin Priority Repair Operator

Sort the vessels in ID_{del} by their corresponding ABQ values in descending order, and insert them into the solution in that order. The insertion process is as follows: first, randomly allocate a certain number of tugboats to assist with the vessel's entry into the port, randomly assign the vessel to one of the available berths at that time, then allocate the maximum number of quay cranes available at that berth at that time (not exceeding the range of quay cranes available to that vessel), calculate the vessel's port processing time and departure time, and randomly allocate a certain number of tugboats to assist with the vessel's departure from the port.

4.4.5. Tugboat Margin Priority Repair Operator

Sort the vessels in ID_{del} by their corresponding ABT values in descending order and insert them into the solution in that order. The insertion process is as follows: allocate the maximum available number of tugboats at that time to assist the vessel's entry into the port, randomly assign the vessel to one of the available berths at that time, then randomly allocate the quay cranes available at that berth at that time (within the vessel's allowable range), calculate the vessel's port processing time and departure time, and allocate the maximum available number of tugboats at that time to assist with the vessel's departure from the port.

4.5. Operator Manager

The operator manager in the ALNS algorithm is one of its core components, responsible for selecting and managing the destruction and repair operators. The design of the operator manager allows the ALNS algorithm to dynamically adjust the frequency and priority of operator usage based on past performance, thereby improving the efficiency and quality of the solution. The following details the key functions and operational mechanisms of the operator manager:

(1) Roulette wheel selection method: The roulette wheel selection method is used by the operator manager to randomly select destruction and repair operators based on their weights. Each operator's weight reflects its historical performance success, with higher-weighted operators having a greater probability of being selected in the future.

(2) Adaptive update: After each iteration, the performance of the operators is evaluated, and their weights are adjusted accordingly based on the improvement or deterioration of the solution. For example, if an operator generates a higher quality solution, its weight is increased; if the result does not improve or worsens, its weight is decreased.

(3) Performance evaluation: Performance evaluation is based on the extent of improvement in the solution generated by the operator. This typically involves comparing the quality of the solution before and after the operator's application such as cost reduction, service level improvement, etc.

(4) Weight adjustment: An exponential smoothing update rule is used for weight adjustment to ensure that the operator's weight reflects its recent performance. This helps the algorithm avoid the pitfall of continually selecting operators that are no longer effective.

4.6. Adaptive Large Neighborhood Search Algorithm Process

The process of the adaptive large neighborhood search (ALNS) algorithm is shown in Algorithm 1.

Algorithm 1. Algorithm execution process.

ALNS

```
Input Parameters:
$T$: Initial temperature
$T_0$: Termination temperature
$qt$: Cooling rate
$I$: Maximum number of iterations
Methods: Set of destruction and repair methods
Weights: Initial weights corresponding to the methods
Start
1. solution ← GenerateInitialSolution ()
2. Set temperature $T\_initial$ to $T$
3. Set cooling rate to $r$
4. Initialize methods and weights
5. Repeat until termination condition is met:
    6. improv ← False
    7. For $i$: = 1 to $I$ Do
        8. destroy_method ← SelectMethod (Methods, Weights)
        9. partial_solution ← Destroy (solution, destroy_method)
        10. repair_method ← SelectMethod (Methods, Weights)
        11. new_solution ← Repair (partial_solution, repair_method)
        12. If Evaluate(new_solution) better than Evaluate(solution) or AcceptByProbability():
            13. solution ← new_solution
            14. UpdateWeights (Methods, Weights, Positive)
            15. improv ← True
        16. Else:
            17. UpdateWeights (Methods, Weights, Negative)
        18. If not improv:
            19. $T \leftarrow T * qt$
        20. If $T < T_0$:
            21. Break
End
Output Solution
```

4.7. Specific Quay Crane Scheduling Solution

After determining a feasible joint allocation plan, specific quay cranes are assigned to each vessel based on this plan, and the specific quay crane scheduling scheme is determined. Since the feasible berth and quay crane allocation plan has already been obtained, only the variables LQ_v and RQ_v remain to be solved. An integer programming method can be used to solve this, convert the solved variables into constants, and remove the constraints related to these already solved variables. The transformed model consists of Equations (1) and (35) to (41). Using CPLEX, the detailed quay crane scheduling scheme can be obtained within 0.1 s.

4.8. Rescheduling of Uncertain Arriving Vessels

When it is determined that the arrival time of certain vessels is delayed, it is necessary to reschedule the corresponding uncertain arriving vessels based on the original scheduling plan. In the rescheduling plan, the schedule of the rescheduled vessels must not conflict with the schedules of other vessels. This insertion process is the same as the solution repair process in the ALNS. The IDs of the vessels that need to be rescheduled are stored in ID_{del}. Therefore, during rescheduling, it is only necessary to call the repair operator in the ALNS algorithm, based on the modified arrival times of the uncertain arriving vessels, to obtain the corresponding rescheduling plan.

5. Case Study Analysis

A port in Xiamen was studied in this paper, where the container handling quay is 1200 m long, equipped with twelve quay cranes and five tugboats. There are three types of incoming vessel: small, medium, and large. The relevant parameters for these three types of vessels are shown in Table 3. The minimum number of tugboats required for different types of ships to enter or leave the port varies. Additionally, the more tugboats allocated to a ship for its entry or departure, the shorter the time needed. Table 4 illustrates the relationship between the time required for various types of ships to enter or leave the port and the number of tugboats assigned to them. For instance, if five tugboats are allocated to a large ship for its entry, the time required for this operation is 1 h. In this paper, time and space were discretized, with the minimum spatial unit being 50 m and the minimum time unit being 1 h. Therefore, the quay was divided into 24 small berths. Additionally, four different scales of examples with 5, 10, 20, and 40 incoming vessels were generated. The planning periods for these four different scales were 60, 120, 180, and 300 h, respectively. The experiments were conducted on a computer equipped with an Intel(R) Core (TM) i5-10400 CPU @ 2.90 GHz processor and 32 GB of RAM, utilizing example programs within the Python framework PyCharm version 2023 and CPLEX solver version 12.6.

Table 3. Parameters of different types of vessels.

Vessel Type	Length (m)	Quay Crane Handling Time (h)	Minimum Required Tugboats	Minimum Required Quay Cranes	Maximum Required Quay Cranes	Proportion (%)
Small	[100, 200]	[10, 50]	1	[2, 2]	[2, 3]	30
Medium	[200, 300]	[20, 80]	2	[2, 3]	[4, 5]	30
Large	[300, 400]	[80, 120]	3	[3, 4]	[6, 7]	40

Table 4. Relationship between vessel entry/exit time and number of tugboats.

Type \ Tugboats	1	2	3	4	5
Small	3	2	2	1	1
Medium	-	3	2	2	1
Large	-	-	3	2	1

"-" indicates that the number of tugboats does not meet the vessel's minimum number requirement.

5.1. Model Validation

To validate the effectiveness of the integrated scheduling model for tugboats, berths, and quay cranes constructed in this paper, a small-scale example with five vessels was randomly generated, and the *GAP* value was set to 1. The specific parameters of the example are shown in Table 5. The example data were imported into CPLEX for solving, and the integrated scheduling solution obtained by CPLEX is shown in Figure 4. This solution satisfies the constraints of tugboats, berths, and quay cranes, and the berthing time intervals between vessels with similar arrival times and the same berth meet the *GAP* reservation length. Therefore, the MILP model constructed in this paper is effective.

To validate the effectiveness of the partial time-variant constraint on the number of quay cranes used in this paper, a time-invariant MILP model was also constructed. In the time-invariant model, the quay cranes allocated to a vessel can only provide loading and unloading services for other vessels after the vessel departs. Five different small-scale examples with five incoming vessels were generated, with the *GAP* value set to 1. The example data were imported into the model for solving, and the results are shown in Table 6. In this table, *OBJ* represents the objective function value and *CT* represents the solving time. It can be observed that the objective function values of the integrated scheduling solutions obtained by considering the partial time-variant quay crane strategy were all less than or equal to those obtained by considering the time-invariant strategy, and the solving times were similar. This proves that the partial time-variant strategy for quay cranes considered in this paper is feasible and effective.

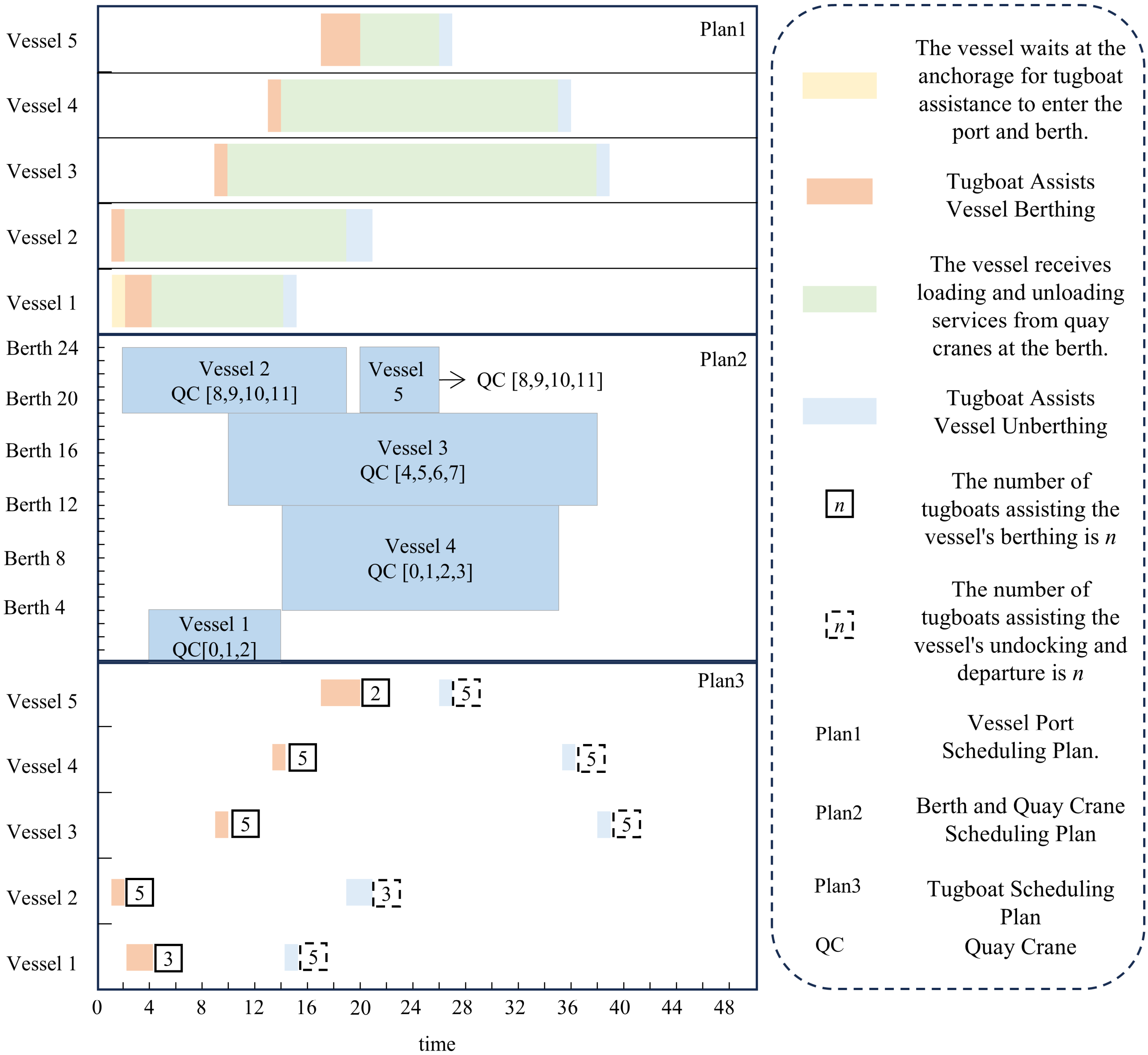


Figure 4. Gantt chart of integrated scheduling for tugboats, berths, and quay cranes.

Table 5. Parameters of small-scale example.

Vessel	Arrival Time	Length	Preferred Berth	Agreed Departure Time	Minimum Quay Cranes	Maximum Quay Cranes	Quay Crane Hours	Vessel Type
1	1	4	4	23	2	3	30	Medium
2	1	5	17	27	2	4	66	Medium
3	9	7	15	35	3	6	109	Large
4	13	8	10	35	3	7	84	Large
5	17	5	19	28	2	4	22	Medium

Table 6. Comparison of different quay crane transfer strategies.

Size	Time-Invariant		Partially Time-Variant	
	OBJ	*CT*	*OBJ*	*CT*
5	2357	115.98	2310	221.23
	2125	105.72	2124	206.47
	1853	5.76	1848	10.12
	1587	16.63	1587	17.46
	1925	5.52	1907	8.85

5.2. Algorithm Validation

5.2.1. Algorithm Parameter Tuning

The outer loop iteration counts I in the adaptive large neighborhood search (ALNS) algorithm was set to 10, based on related research. The inner loop iteration count is determined by T, T_0, and qt. Let T_0 and qt be 1 and 0.99, respectively. By using different scale examples, we can find the initial temperature T that results in a low objective function value and short runtime. Then, randomly generate four different scale examples with five, ten, twenty, and forty vessels, set different T values for solving, and calculate the objective function value *OBJ* and computation time *CT*, which are the averages of five runs under the corresponding conditions. Perform a quadratic fit of the results, as shown in Figure 5. The upper and lower limits of the curve on the vertical axis represent the maximum and minimum values obtained by the algorithm. Analysis indicates that the optimal T values for the four different scales of examples with 5, 10, 20, and 40 vessels should be 2000, 1000, 1000, and 500, respectively.

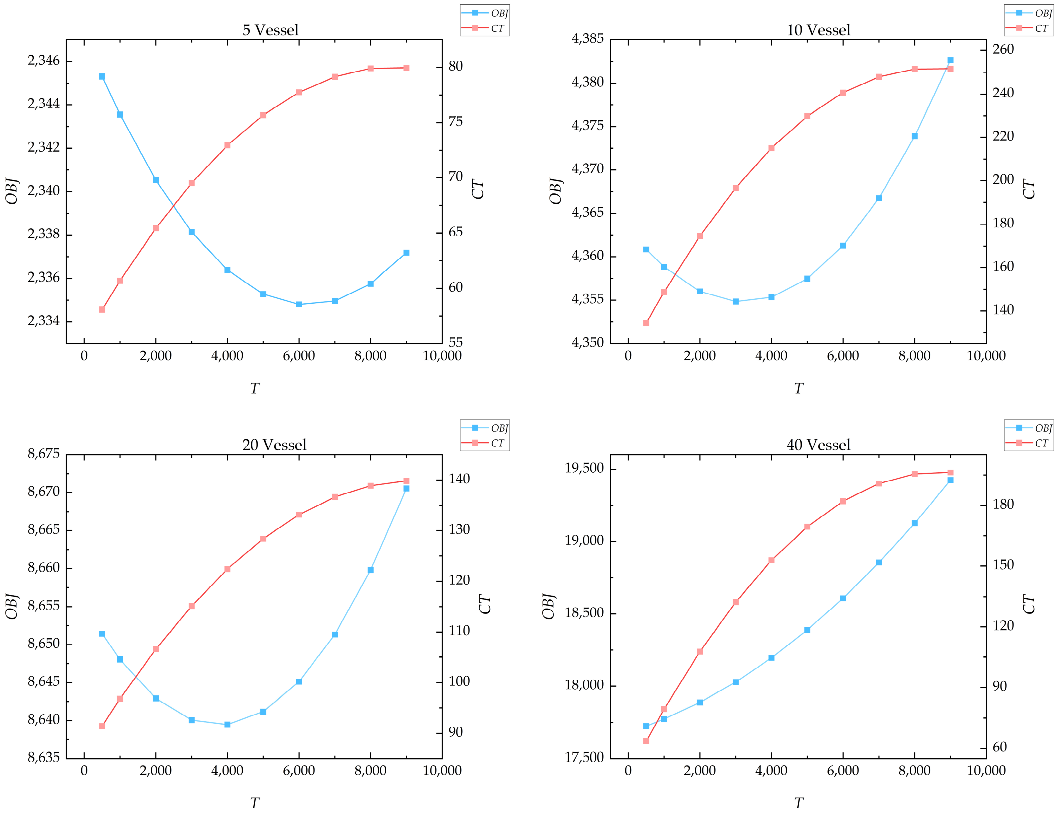


Figure 5. Comparison of algorithm parameters for different scale examples.

5.2.2. Algorithm Comparison Analysis

To verify the effectiveness of the ALNS algorithm designed in this paper, four different scales of examples were randomly generated based on the port's vessel arrival patterns, with five examples generated for each scale. These examples of small-scale were solved using both ALNS and CPLEX, and the results are shown in Table 7. The analysis indicates that for the small-scale instances, CPLEX can obtain the optimal tugboat–berth–quay crane integrated scheduling solution within 5 h, while ALNS can achieve a feasible solution with an average deviation of 0.72% from the optimal solution within 2 min, proving the effectiveness of the proposed algorithm. Moreover, as the size of the test cases gradually increased, the CPLEX solving time escalated significantly. When the example scale increased to more than nine vessels, CPLEX failed to find the integrated scheduling solution within 5 h, while ALNS was less affected by the increase in scale and could obtain a feasible integrated scheduling solution within 3 min. In summary, the ALNS algorithm designed in this paper is effective, has high solving efficiency, and is more suitable for solving large-scale tugboat–berth–quay crane integrated scheduling problems for larger numbers of incoming vessels.

Table 7. Comparison of ALNS and CPLEX solutions of small-scale instance.

Size	ALNS							CPLEX		Gap /%
	OBJ	F_1	F_2	F_3	F_4	F_5	*CT*	*OBJ*	*CT*	
5	2327	210	90	12	240	1775	63.82	2310	221.23	0.73
	2149	194	80	25	260	1590	75.77	2124	206.47	1.17
	1848	144	0	4	235	1465	12.86	1848	10.12	0
	1588	172	10	1	195	1210	24.47	1587	17.46	0.06
	1907	148	0	9	235	1515	35.47	1907	8.85	0
6	2543	242	100	16	275	1910	59.74	2523	722.87	0.79
	2526	246	150	20	265	1845	66.97	2493	213.04	1.32
	2645	244	180	36	275	1910	57.68	2607	303.16	1.45
	2141	202	0	24	285	1630	51.56	2130	38.42	0.51
	2201	200	60	26	285	1630	56.05	2194	331.75	0.31
7	2710	250	120	20	320	2000	67.86	2678	794.59	1.19
	2807	270	160	27	325	2025	64.09	2782	1186.87	0.89
	3444	296	270	48	355	2475	80.30	3410	478.75	0.99
	2911	244	130	27	310	220	56.12	2892	110.85	0.65
	3003	240	200	28	330	2205	58.37	2970	382.05	1.11
8	2931	260	140	21	370	2140	59.66	2884	10,295.84	1.62
	3038	294	160	39	385	2160	66.54	3011	13,966.44	0.92
	4296	348	470	53	405	3020	85.37	4284	12,785.59	0.28
	3577	276	160	31	370	2740	73.89	3569	135.97	0.22
	3702	288	230	39	410	2735	83.09	3674	9541.64	0.76
9	3684	308	190	31	430	2725	64.70	-	-	-
	3700	334	160	41	430	2735	66.08	-	-	-
	4623	382	590	56	445	3150	101.53	-	-	-
	4032	322	160	45	440	3065	71.78	3999	4229.63	0.82
	4144	326	270	43	435	3070	81.58	-	-	-

'-' represents that the optimal solution could not be found within 5 h. Gap = [OBJ(ALNS) − OBJ(CPLEX)]/OBJ(CPLEX) × 100%.

To validate the superiority of the proposed algorithm, four different scales of examples were randomly generated, with five examples generated for each scale. These examples were solved using ALNS, large neighborhood search (LNS), variable neighborhood search (VNS), and genetic algorithm (GA), respectively. The explanations regarding VNS, LNS, and GA are presented in Appendix A. The specific GA operation process can be found in Appendix B. The results are shown in Table 8. Analysis indicates that the solution quality of ALNS was higher than that of other neighborhood search algorithms (LNS and VNS) and the swarm intelligence algorithm (GA). Additionally, neighborhood search heuristics are less affected by the increase in example scale, whereas swarm intelligence heuristics experience a significant increase in solving time due to the complexity of the problem and the limitation on the number of solutions. Therefore, compared to other algorithms, the ALNS has more advantages in solving the integrated scheduling of tugboats, berths, and quay cranes.

Table 8. Comparison of solutions from different algorithms.

Size	ALNS		LNS		VNS		GA	
	OBJ	***CT***	***OBJ***	***CT***	***OBJ***	***CT***	***OBJ***	***CT***
5	2327	63.82	2400	76.334	2389	90.41	2426	172.10
	2149	75.77	2321	64.95	2155	114.65	2210	156.78
	1848	12.86	1860	10.87	1849	12.75	1904	143.21
	1588	24.47	1590	13.22	1590	18.22	1654	105.92
	1907	35.47	1909	19.19	1918	25.29	1977	123.45
10	4211	61.14	4292	102.08	4348	82.03	4674	412.34
	4318	102.28	4474	70.13	4618	154.68	5008	347.56
	4851	109.82	4941	61.66	5192	221.52	5530	468.78
	4337	59.19	4568	98.17	4591	151.77	5074	385.21
	4663	64.45	4934	67.95	4737	61.41	5520	492.90
20	8186	84.07	8453	97.25	8456	161.75	9659	987.65
	8176	122.59	8199	90.91	8280	94.32	9892	865.78
	8557	114.73	8855	127.54	8797	116.90	10,182	934.21
	8462	121.39	8844	72.90	8647	74.34	9589	1078.90
	8785	134.50	8811	89.06	8828	132.98	10,366	823.45
40	18,158	119.16	18,601	212.96	18,610	181.15	21,426	1956.34
	17,597	178.71	17,737	108.90	17,846	190.13	21,292	2078.92
	18,598	190.20	19,821	146.13	20,259	172.186	23,433	2345.67
	18,956	195.78	20,414	164.04	19,318	135.55	24,074	2123.89
	18,591	213.57	20,941	248.40	18,643	186.79	22,495	2290.10

5.3. Sensitivity Analysis

To validate the impact of uncertainty factors and berth preference on integrated scheduling, different buffer time intervals (*GAP*) and berth deviation control coefficients (C_3) were set. A case with 20 arriving vessels was used as the research object, and the objective function value (*OBJ*) as well as F_1, F_2, F_3, F_4, and F_5 under different *GAP* and C_3 conditions were calculated. The results are shown in Figure 6. Analysis showed that as C_3 increased, *OBJ* also increased. The increase in *OBJ* was mainly related to the increase in F_1, F_2, and F_3. As C_3 increased, the rate of increase in F_3 gradually decreased, indicating that the berth deviation of all vessels became smaller as C_3 increased. Meanwhile, the combined increase in F_1 and F_2 showed an increasing trend, indicating that an increase in C_3 led to longer vessel waiting times in the port, delayed departure times, and increased time in the port. It is necessary to balance the berth deviation costs and time-related costs based on the actual operating conditions of the port and select an appropriate C_3 value.

An increase in the vessel buffer time *GAP* led to an increase in F_1, F_2 and *OBJ*, while the values of F_3, F_4, and F_5 remained relatively unchanged, resulting in an increased *OBJ* value. Reserving a certain buffer time interval causes vessels with similar arrival times and berths to wait for a period before entering the port for loading and unloading services, thereby delaying the berthing and departure times and increasing the waiting and delayed departure costs. When determining the situation of uncertain arriving vessels, it is necessary to reschedule the uncertain vessels based on the original scheduling plan. After rescheduling, the scheduling cost will inevitably increase. The higher the stability of the original schedule, the smaller the increase in cost compared to the original plan after rescheduling. The increase in rescheduling cost is denoted as R_c, where R_c = OBJ^* −

OBJ, with *OBJ** being the total cost after rescheduling and OBJ being the total cost of the original schedule. To verify the stability of the model under different *GAP* values, this study introduced various scenarios of uncertain vessel arrivals. The key metrics for measuring these uncertain scenarios were the number of delayed arriving vessels, denoted as θ, and the vessel arrival delay time, denoted as β. Figure 7 shows the R_c values obtained by the ALNS algorithm under different θ, β, and *GAP* conditions for a scenario with a fleet of 20 vessels. The analysis of Figure 7 revealed that the larger the *GAP* value, the smaller the R_C value of the obtained plan, indicating higher model stability. Therefore, an appropriate *GAP* value should be selected based on the uncertainty fluctuation of vessel arrival times at the port, rather than arbitrarily choosing a large buffer time interval to cope with the uncertainty of vessel arrival times.

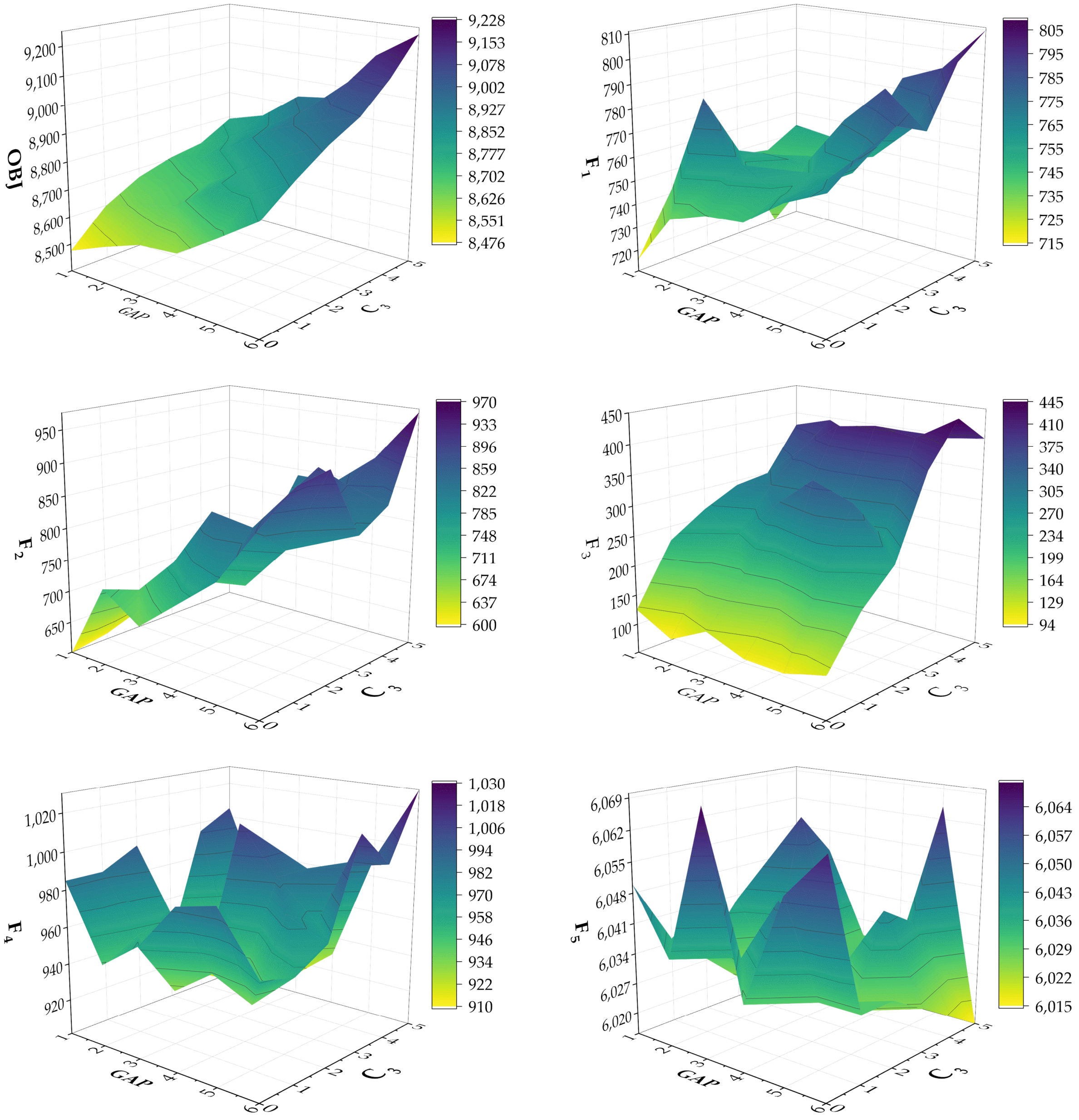


Figure 6. Sensitivity analysis 3D plot.

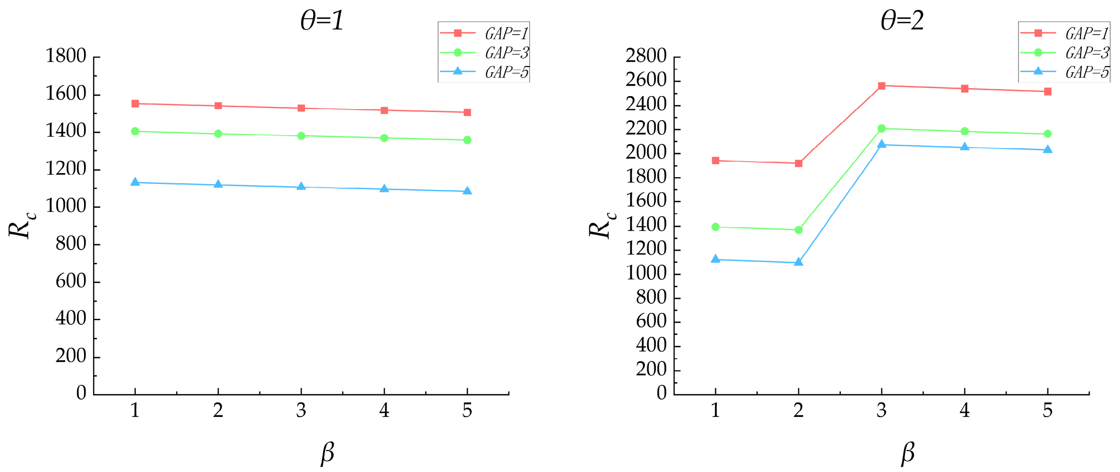


Figure 7. The relationship between stability and *GAP*.

6. Conclusions

To address the increasing challenges of maritime demand at container ports and considering the limited scheduling resources, this study investigated the integrated scheduling of port resources such as tugboats, berths, and quay cranes to provide efficient scheduling solutions for ports. The research focused on integrated scheduling under conditions of uncertain vessel arrival times, vessel berthing preferences, and the constraint that quay cranes cannot cross each other. A mixed-integer linear programming (MILP) model was developed with the objective of minimizing the total costs including vessel fuel consumption, vessel waiting and departure delay costs, berthing deviation costs, quay crane handling costs, and tugboat fuel costs. An adaptive large neighborhood search (ALNS) algorithm was designed with destruction and repair operators specific to the characteristics of the integrated scheduling problem, enhancing the search efficiency and solution quality of the algorithm. The conclusions drawn from the data experiments are as follows:

(1) The constructed MILP model is effective, providing optimal integrated scheduling solutions for tugboats, berths, and quay cranes in a short time for small-scale instances. Considering certain time-varying strategies can yield better scheduling solutions.

(2) Compared to CPLEX, the ALNS algorithm can find optimal or near-optimal solutions in a shorter time for small-scale instances. For larger-scale instances where CPLEX could not find the optimal solution, ALNS still provided high-quality scheduling solutions efficiently. Compared to other algorithms such as LNS, VNS, and GA, ALNS delivers higher-quality solutions.

(3) The scheduling cost increases as the buffer time interval (*GAP*) between vessels increases. The *GAP* value should be set based on the uncertainty level of vessel arrival times.

(4) A larger C_3 value reduces the degree of berthing deviation, ensuring that all vessels are serviced near their preferred berths, but it also increases the scheduling cost.

Future research on port landside scheduling can extend from the integration of tugboats, berths, and quay cranes to include other components such as yard cranes, achieving integrated port scheduling decisions. In terms of algorithm research, more efficient algorithms or machine learning methods can be used for scheduling studies.

Author Contributions: Conceptualization, L.C., J.Z. and Q.Y.; Methodology, L.C. and J.Z.; Software, J.Z. and Q.Y.; Validation J.Z. and X.C.; Formal analysis J.Z., Q.Y. and X.C.; Investigation, L.C. and J.Z.; Resources, L.C. and J.Z.; Data curation, J.Z., Q.Y. and X.C.; Writing—original draft preparation, L.C. and J.Z.; Writing—review and editing, L.C., J.Z. and Q.Y.; Visualization, J.Z., Q.Y. and X.C.; Supervision, L.C. and Q.Y.; Project administration, L.C.; Funding acquisition, L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Social Science Fund of China [grant number 23&ZD138].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data, models, or code generated or used during the study are available from the corresponding author by request.

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

Appendix A

Method	Explanation
VNS	The variable neighborhood search (VNS) is a metaheuristic algorithm based on local search, designed to solve combinatorial optimization problems. It systematically changes the neighborhood structure of solutions to escape local optima, thereby exploring a broader solution space. The core idea of VNS is to dynamically adjust and switch between different neighborhood structures during the search process, gradually finding better so-lutions.
LNS	The large neighborhood search (LNS) is a metaheuristic algorithm used to solve complex optimization problems. It explores a broader solution space by perturbing a larger portion of the solution in each iteration to find better solutions. The core idea of LNS is to selectively destroy and reconstruct the current solution to escape local optima and search for the global optimum.
GA	The genetic algorithm (GA) is an optimization algorithm based on the principles of natural selection and ge-netics. It simulates the process of biological evolution, iteratively optimizing candidate solutions through op-erations such as selection, crossover, and mutation. Genetic algorithms are particularly well-suited for prob-lems with large search spaces, complex objective functions, or those that are difficult to differentiate. By con-tinuously iterating, they can find approximate optimal solutions to the problem.

Appendix B

(1) Solution encoding: As shown in Figure A1, the solution encoding adopts a four-layer encoding structure.

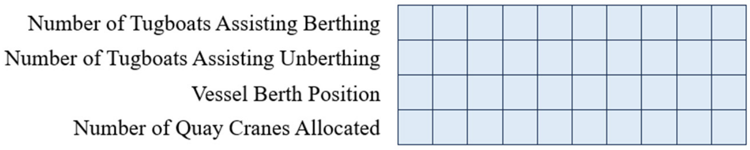

Number of Tugboats Assisting Berthing
Number of Tugboats Assisting Unberthing
Vessel Berth Position
Number of Quay Cranes Allocated

Figure A1. Solution Encoding Method.

(2) Crossover operation: Figure A2 illustrates the method of solution crossover.

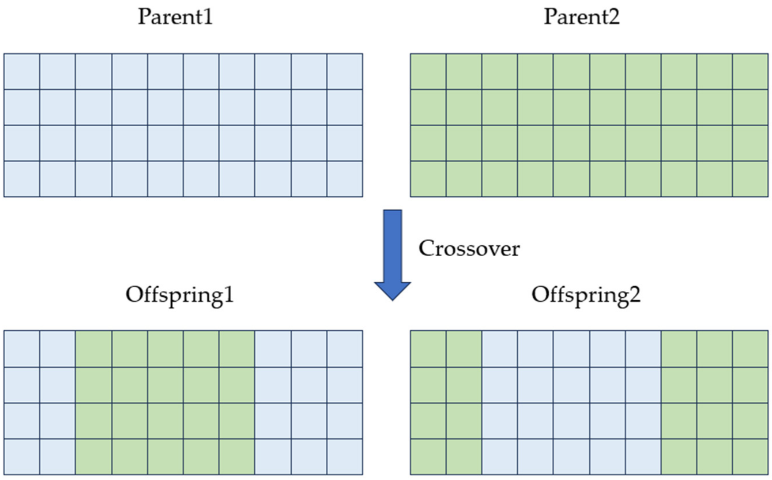


Figure A2. Solution crossover.

(3) Mutation operation: Figure A3 illustrates the mutation operation of the solution.

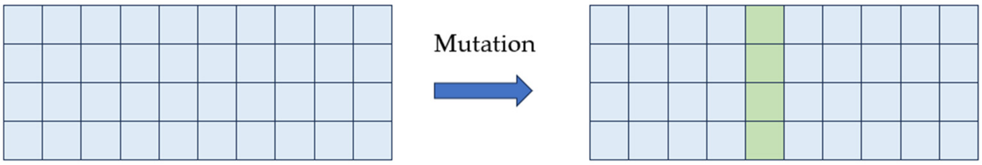


Figure A3. Solution mutation.

(4) Solution repair operation: Figure A4 illustrates the solution repair operation. When calculating the objective function value, the solution must first undergo repair. The repair process involves using a heuristic designed based on the FCFS (first-come, first-served) principle to repair the solution. Once the repair is completed, the objective function value can be calculated.

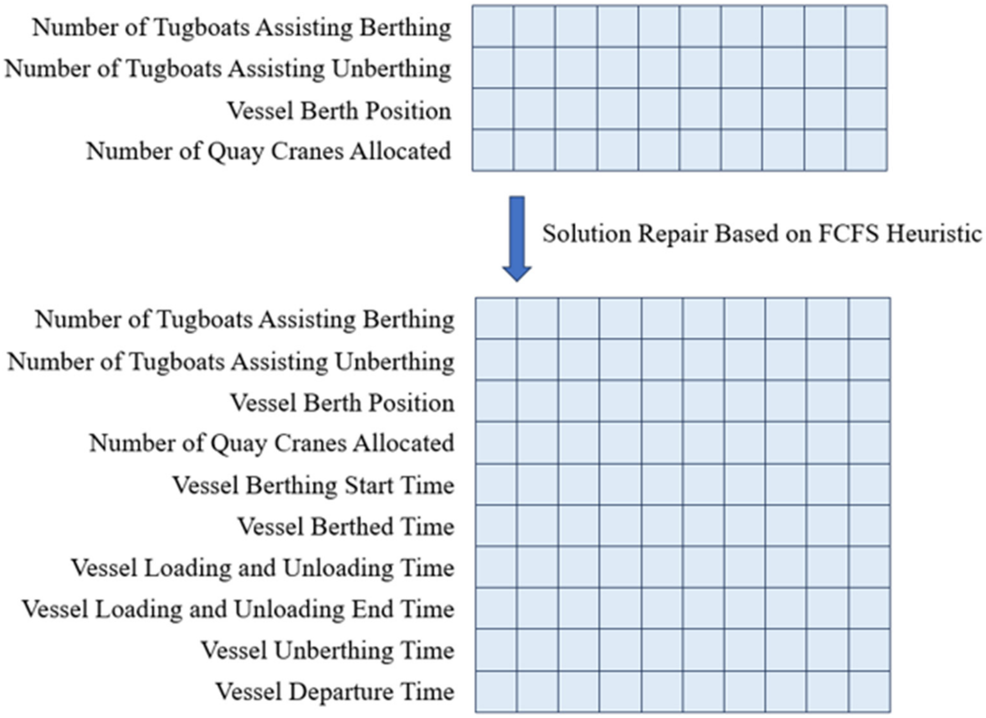

Number of Tugboats Assisting Berthing
Number of Tugboats Assisting Unberthing
Vessel Berth Position
Number of Quay Cranes Allocated
Solution Repair Based on FCFS Heuristic
Number of Tugboats Assisting Berthing
Number of Tugboats Assisting Unberthing
Vessel Berth Position
Number of Quay Cranes Allocated
Vessel Berthing Start Time
Vessel Berthed Time
Vessel Loading and Unloading Time
Vessel Loading and Unloading End Time
Vessel Unberthing Time
Vessel Departure Time

Figure A4. Solution repair.

References

1. Imai, A.; Nishimura, E.; Papadimitriou, S. The Dynamic Berth Allocation Problem for a Container Port. *Transp. Res. Part. B Methodol.* **2001**, *35*, 401–417. [CrossRef]
2. Prencipe, L.P.; Marinelli, M. A Novel Mathematical Formulation for Solving the Dynamic and Discrete Berth Allocation Problem by Using the Bee Colony Optimisation Algorithm. *Appl. Intell.* **2021**, *51*, 4127–4142. [CrossRef]
3. Iris, Ç.; Lalla-Ruiz, E.; Lam, J.S.L.; Voß, S. Mathematical Programming Formulations for the Strategic Berth Template Problem. *Comput. Ind. Eng.* **2018**, *124*, 167–179. [CrossRef]
4. Umang, N.; Bierlaire, M.; Vacca, I. Exact and Heuristic Methods to Solve the Berth Allocation Problem in Bulk Ports. *Transp. Res. Part E Logist. Transp. Rev.* **2013**, *54*, 14–31. [CrossRef]
5. 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]
6. Venturini, G.; Iris, Ç.; Kontovas, C.A.; Larsen, A. The Multi-Port Berth Allocation Problem with Speed Optimization and Emission Considerations. *Transp. Res. Part D Transp. Environ.* **2017**, *54*, 142–159. [CrossRef]

7. Ernst, A.T.; Oğuz, C.; Singh, G.; Taherkhani, G. Mathematical Models for the Berth Allocation Problem in Dry Bulk Terminals. *J. Sched.* **2017**, *20*, 459–473. [CrossRef]
8. Cheimanoff, N.; Fontane, F.; Kitri, M.N.; Tchernev, N. A Reduced VNS Based Approach for the Dynamic Continuous Berth Allocation Problem in Bulk Terminals with Tidal Constraints. *Expert Syst. Appl.* **2021**, *168*, 114215. [CrossRef]
9. Martin-Iradi, B.; Pacino, D.; Ropke, S. An Adaptive Large Neighborhood Search Heuristic for the Multi-Port Continuous Berth Allocation Problem. *Eur. J. Oper. Res.* **2024**, *316*, 152–167. [CrossRef]
10. Xiang, X.; Liu, C. An Almost Robust Optimization Model for Integrated Berth Allocation and Quay Crane Assignment Problem. *Omega* **2021**, *104*, 102455. [CrossRef]
11. Wang, Z.; Cheng, J.; Hu, H. A Proactive-Reactive-Based Approach for Continuous Berth Allocation and Quay Crane Assignment Problems with Hybrid Uncertainty. *J. Mar. Sci. Eng.* **2024**, *12*, 182. [CrossRef]
12. Ji, B.; Huang, H.; Yu, S.S. An Enhanced NSGA-II for Solving Berth Allocation and Quay Crane Assignment Problem With Stochastic Arrival Times. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 459–473. [CrossRef]
13. Liu, W.; Zhu, X.; Wang, L.; Li, S. Rolling Horizon Based Robust Optimization Method of Quayside Operations in Maritime Container Ports. *Ocean Eng.* **2022**, *256*, 111505. [CrossRef]
14. Chargui, K.; Zouadi, T.; Sreedharan, V.R.; El Fallahi, A.; Reghioui, M. A Novel Robust Exact Decomposition Algorithm for Berth and Quay Crane Allocation and Scheduling Problem Considering Uncertainty and Energy Efficiency. *Omega* **2023**, *118*, 102868. [CrossRef]
15. Yu, J.; Tang, G.; Voß, S.; Song, X. Berth Allocation and Quay Crane Assignment Considering the Adoption of Different Green Technologies. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *176*, 103185. [CrossRef]
16. Li, Y.; Chu, F.; Zheng, F.; Liu, M. A Bi-Objective Optimization for Integrated Berth Allocation and Quay Crane Assignment With Preventive Maintenance Activities. *IEEE Trans. Intell. Transport. Syst.* **2022**, *23*, 2938–2955. [CrossRef]
17. Iris, Ç.; Pacino, D.; Ropke, S.; Larsen, A. Integrated Berth Allocation and Quay Crane Assignment Problem: Set Partitioning Models and Computational Results. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *81*, 75–97. [CrossRef]
18. Malekahmadi, A.; Alinaghian, M.; Hejazi, S.R.; Assl Saidipour, M.A. Integrated Continuous Berth Allocation and Quay Crane Assignment and Scheduling Problem with Time-Dependent Physical Constraints in Container Terminals. *Comput. Ind. Eng.* **2020**, *147*, 106672. [CrossRef]
19. Krimi, I.; Benmansour, R.; El Cadi, A.A.; Deshayes, L.; Duvivier, D.; Elhachemi, N. A Rolling Horizon Approach for the Integrated Multi-Quays Berth Allocation and Crane Assignment Problem for Bulk Ports. *Int. J. Ind. Eng. Comput.* **2019**, *10*, 577–591. [CrossRef]
20. Iris, Ç.; Lam, J.S.L. Recoverable Robustness in Weekly Berth and Quay Crane Planning. *Transp. Res. Part B Methodol.* **2019**, *122*, 365–389. [CrossRef]
21. He, J.; Huang, Y.; Yan, W.; Wang, S. Integrated Internal Truck, Yard Crane and Quay Crane Scheduling in a Container Terminal Considering Energy Consumption. *Expert Syst. Appl.* **2015**, *42*, 2464–2487. [CrossRef]
22. Wang, T.; Wang, X.; Meng, Q. Joint Berth Allocation and Quay Crane Assignment under Different Carbon Taxation Policies. *Transp. Res. Part B Methodol.* **2018**, *117*, 18–36. [CrossRef]
23. Cheimanoff, N.; Fontane, F.; Kitri, M.N.; Tchernev, N. Exact and Heuristic Methods for the Integrated Berth Allocation and Specific Time-Invariant Quay Crane Assignment Problems. *Comput. Oper. Res.* **2022**, *141*, 105695. [CrossRef]
24. Iris, Ç.; Pacino, D.; Ropke, S. Improved Formulations and an Adaptive Large Neighborhood Search Heuristic for the Integrated Berth Allocation and Quay Crane Assignment Problem. *Transp. Res. Part E Logist. Transp. Rev.* **2017**, *105*, 123–147. [CrossRef]
25. Xu, Q.; Mao, J.; Jin, Z. Simulated Annealing-Based Ant Colony Algorithm for Tugboat Scheduling Optimization. *Math. Probl. Eng.* **2012**, *2012*, 246978. [CrossRef]
26. Wei, X.; Meng, Q.; Lim, A.; Jia, S. Dynamic Tugboat Scheduling for Container Ports. *Marit. Policy Manag.* **2023**, *50*, 492–514. [CrossRef]
27. Sun, C.; Li, M.; Chen, L.; Chen, P. Dynamic Tugboat Scheduling for Large Seaports with Multiple Terminals. *J. Mar. Sci. Eng.* **2024**, *12*, 170. [CrossRef]
28. Li, J.; Duan, X.; Xiong, Z.; Yao, P. Tugboat Scheduling Method Based on the NRPER-DDPG Algorithm: An Integrated DDPG Algorithm with Prioritized Experience Replay and Noise Reduction. *Sustainability* **2024**, *16*, 3379. [CrossRef]
29. Zhong, H.; Zhang, Y.; Gu, Y. A Bi-Objective Green Tugboat Scheduling Problem with the Tidal Port Time Windows. *Transp. Res. Part D Transp. Environ.* **2022**, *110*, 103409. [CrossRef]
30. Kang, L.; Meng, Q.; Tan, K.C. Tugboat Scheduling under Ship Arrival and Tugging Process Time Uncertainty. *Transp. Res. Part E-Logist. Transp. Rev.* **2020**, *144*, 102125. [CrossRef]
31. Wei, X.; Jia, S.; Meng, Q.; Tan, K.C. Tugboat Scheduling for Container Ports. *Transp. Res. Part E-Logist. Transp. Rev.* **2020**, *142*, 102071. [CrossRef]
32. Jia, S.; Li, S.; Lin, X.; Chen, X. Scheduling Tugboats in a Seaport. *Transp. Sci.* **2021**, *55*, 1370–1391. [CrossRef]

33. Wang, X.; Liang, Y.; Wei, X.; Chew, E.P. An Adaptive Large Neighborhood Search Algorithm for the Tugboat Scheduling Problem. *Comput. Ind. Eng.* **2023**, *177*, 109039. [CrossRef]
34. Yang, Z.-Y.; Cao, X.; Xu, R.-Z.; Hong, W.-C.; Sun, S.-L. Applications of Chaotic Quantum Adaptive Satin Bower Bird Optimizer Algorithm in Berth-Tugboat-Quay Crane Allocation Optimization. *Expert Syst. Appl.* **2024**, *237*, 121471. [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.