

Article

Joint Port–Lock Scheduling Optimization Under the Direct Transshipment Mode Between Seagoing and River Vessels

Shuaiqi Wang ^{1,*}, Yong Zhang ^{1,*}, Jian Li ², Jiayi Shi ², Jiashan Yuan ¹ and Ying Yang ²¹ School of Transportation, Southeast University, Nanjing 211189, China; 230228906@seu.edu.cn (S.W.)² College of Information Management, Nanjing Agricultural University, Nanjing 211800, China; lijianzh@njau.edu.cn (J.L.)

* Correspondence: zhangyong@seu.edu.cn

Abstract

This study addresses the problems of vessel congestion and inefficient resource utilization caused by the independent scheduling of port berths and locks in river–sea intermodal transport. Moreover, this study explores the joint scheduling optimization problem for a single port and a single lock under the direct transshipment mode of seagoing and river vessels. First, considering the spatiotemporal coupling characteristics of the port–lock system, a mixed-integer linear programming model integrating seaside and riverside berth allocation, lock batch scheduling, and 2D layout constraints of the lock chambers is constructed. The objective is to minimize the time costs of seagoing vessels in a port, river vessels in a port, and river vessels passing through the locks. Second, a hierarchical iterative collaborative optimization algorithm is designed. The outer layer uses an iterative algorithm framework, while the inner layer utilizes a heuristic algorithm to address the lock scheduling problem. An adaptive large neighborhood search algorithm is used to address the berth scheduling problem. Finally, the model and algorithm are validated through numerical instances. Compared with the commercial solver CPLEX, the proposed algorithm HICOA achieves up to 81.45% lower costs in medium-to-large instances. Compared with independent scheduling, joint scheduling reduces the total vessel dwell time cost by up to 66.22%, demonstrating the significant value of port–lock collaboration. Furthermore, the arrival time window of river vessels and the ratio of the number of vessels to the number of berths have a significant influence on system performance.

Keywords: river–sea intermodal transport; direct transshipment mode; joint scheduling of port and locks; hierarchical iterative collaborative optimization algorithm; adaptive large neighborhood search algorithm

Academic Editor: Anna Sciomachen

Received: 19 June 2026

Revised: 20 July 2026

Accepted: 21 July 2026

Published: 22 July 2026

MSC: 90B35; 90C06; 90C59; 90B06

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

In the global waterway network, the “port–lock” system, consisting of locks and adjacent ports, is the primary hub of river–sea intermodal transport. Port berth allocation and lock scheduling have been largely independently optimized for a long time. This lack of global coordination easily leads to a vicious cycle of “waiting for berths to pass through locks” and “waiting for lock times to berth”, resulting in severe vessel congestion and resource idleness, which has become a major bottleneck restricting waterway efficiency

(Zhang Y. et al., 2023) [1]. Currently, China is vigorously promoting the construction of the Pinglu Canal in Guangxi, an important project. The operation of the Pinglu Canal, a key channel connecting the Xijiang River inland waterway and the Beibu Gulf seaports, will heavily rely on efficient “port–lock” coordination. In the context of direct transshipment between seagoing vessels and river vessels, traditional fragmented scheduling is unlikely to satisfy the continuity and high-efficiency requirements of vessel navigation. Therefore, for emerging canal systems, such as Pinglu Canal, breaking down business barriers and conducting research on port–lock joint scheduling optimization under the direct transshipment model of seagoing vessels and river vessels is of great practical significance for maximizing the efficiency of the new canal and fundamentally avoiding navigation congestion.

In the complex scenario of direct transfer between seagoing and river vessels, the key to improving the efficiency of port and lock scheduling through joint optimization lies in the stringent spatiotemporal coupling characteristics and resource dependency relationships between the two operations. Traditional single-vessel berthing operations have been upgraded to synchronous coordination between river and seagoing vessels. The direct transfer time window of a vessel in the port and its preset lock passage time window present a deeply bound two-way constraint relationship. If port and lock scheduling decisions lack coordination and are disconnected, then two typical problems are likely to occur. First, when the lockage frequency is high, ships can pass through the lock area without waiting. However, berthing waiting may occur due to untimely matching of port berth resources. Second, when the lockage frequency is relatively loose, port berths can avoid ship queuing and congestion. Nevertheless, ship congestion and waiting may occur in the lock area. Both situations reduce system operating efficiency. Meanwhile, if lock frequency planning is integrated with port berth allocation and loading/unloading resource configuration, then it can effectively guide the orderly flow and smooth connection of ships between the port and locks. The entire process time of “arrival–operation–departure” and the passage time of “arrival at the lock entry into the lock exit from the lock” can be accurately matched through a collaborative optimization mechanism, thereby reducing the ineffective dwell time of ships at anchorages and in front of the locks. Accordingly, the utilization rates of port berths and lock chambers are improved, and the overall flow efficiency of the shipping system is maximized.

Although berth allocation and lock scheduling have been extensively studied, the exploration of the substantial joint optimization of the two is still in its developmental stage. Existing research can be mainly divided into two categories: one focuses on the resource set within the port; the other focuses on the scheduling optimization within the lock. A few pioneering studies have begun to examine port–lock coordination. Ji et al. (2018) [2] treated the lock and terminal as an integrated system for coordinated scheduling. Liu et al. (2024) [3] studied the coordination of waterways and locks during the flood season. However, most existing joint scheduling models focus on specific links, such as only considering riverside transshipment, and lack port–lock joint scheduling in the context of river–sea intermodal transport with direct transshipment between seagoing vessels and river vessels. In this port–lock joint system, river vessels pass through the lock downstream, reach the berth, dock with seagoing vessels to complete loading and unloading, and sail upstream to leave the lock.

The primary problem addressed in this study is the port–lock joint scheduling optimization of the direct transshipment mode between seagoing vessels and river vessels against the above-mentioned river–sea intermodal transport background. The academic difficulties are mainly reflected in three aspects:

- (1) High model complexity. The problem requires the integration of discrete berth allocation, continuous operation time arrangement, the batch processing of lock scheduling, and operation connection constraints between seagoing vessels and river vessels at the same time, resulting in numerous model variables and complex constraints, constituting a strongly coupled mixed-integer programming problem.
- (2) Significant challenge in solving. The combinatorial explosion characteristic of the problem makes it an NP-hard problem. Exact algorithms have difficulty coping with the actual scale, and an efficient heuristic or metaheuristic algorithm that can simultaneously handle the berth sequence, operation time sequence, lock allocation, and cabin layout is extremely challenging to design.
- (3) Close spatiotemporal correlation. A natural conflict exists between the batch processing characteristics of locks and the continuous time operation of ports. Coordination of the time windows of the two requires careful modeling to ensure the feasibility and optimality of the solution.

In light of the above-mentioned difficulties and shortcomings, the main contributions of this study are as follows:

- (1) A new joint optimization mixed-integer linear programming model integrating sea-side berth allocation, riverside berth allocation, lock batch scheduling, and lock chamber 2D layout constraints is constructed, which comprehensively describes the actual operation of the river–sea intermodal transport system.
- (2) Given the characteristics of strong model complexity and difficulty in solving, this study proposes a hierarchical iterative collaborative optimization algorithm (HICOA) for the characteristics of this port–lock system. Specifically, an iterative algorithm is used as the overall framework, and a heuristic algorithm is designed to solve the lock scheduling. Moreover, an adaptive large neighborhood search is utilized to address the berth scheduling of seagoing vessels and river-going ships in the port, thereby obtaining a high-quality feasible solution within a reasonable time.
- (3) The efficiency improvement potential of joint scheduling compared with traditional independent scheduling is quantitatively evaluated through numerical experiments and a case analysis, providing decision support for the collaborative operation of “port–lock”.

The remainder of this paper is organized as follows. Section 2 provides a systematic review of relevant literature; Section 3 elaborates on the problem and establishes a mathematical model; Section 4 introduces the design of the solution algorithm; Section 5 presents numerical experiments and result analysis; Section 6 summarizes the entire work and introduces future research directions.

2. Literature Review

Various scholars around the world have extensively researched berth allocation and lock scheduling, gradually forming a research system from single resource optimization to system-level coordination. This study systematically reviews the relevant research progress in three dimensions: the berth allocation problem, lock scheduling problem, and joint scheduling problem of berths and locks.

2.1. Berth Allocation Problem

Berth allocation is a fundamental scheduling task in port operations. Its objective is to allocate optimal berth locations and operating times to ships based on their arrival time, size, and loading/unloading requirements to minimize the total service time, waiting time, or port operating costs. This problem can be divided into discrete and continuous types

according to berth layout, and both have been proven to be NP-hard problems, requiring heuristic algorithms to balance accuracy and efficiency.

Early research mainly focused on the accurate modeling and solution of single-type berth allocation problems. Cordeau et al. (2005) [4] proposed a time-based tabu search heuristic and a continuous extended heuristic that considers spatiotemporal constraints for discrete dynamic berth allocation problems. Verification based on actual data from the port of Gioia Tauro demonstrated the efficiency of this method, laying the algorithmic foundation for subsequent research. Research is gradually moving toward integration as port operations become increasingly complex. Agra and Oliveira (2018) [5] integrated berth allocation with quay crane scheduling by developing a mixed-integer programming model based on the relative position formula and a discretized improved model. Combined with a rolling time-domain heuristic algorithm, this approach can obtain optimal solutions for real-scale instances with a short period of time, thereby addressing the problem of “weak lower bound and slow solution” in traditional models.

The optimization of berth allocation in special scenarios has become a major research focus in recent years. Correcher et al. (2019) [6] used a mixed-integer programming model for irregularly laid-out wharves to represent spatial constraints, such as adjacency and congestion, and proposed an iterative local search algorithm. Corry and Bierwirth (2019) [7] coupled berth allocation with channel scheduling to address channel–berth connection constraints. Moreover, they developed a waiting-free bidirectional flow workshop model and proposed a corresponding decomposition method to minimize suboptimal allocation caused by channel congestion. Jia et al. (2020) [8] proposed using a three-stage simulation optimization method for dynamic scenarios to address the collaborative needs of deep-sea and feeder vessels. Korekane et al. (2024) [9] designed a neural network-assisted branch and bound algorithm to improve the solution efficiency of dynamic berth allocation problems. Wang et al. addressed the needs of continuous berth allocation and the dynamic scheduling of quay cranes in container ports by formulating the problem as a Markov decision process. They developed a deep reinforcement learning algorithm for near-end policy optimization combined with an A2C neural network framework, adapting to dynamic scheduling environments. Other studies addressing uncertainty in ship arrival times have used the LightGBM model to predict arrival times and integrated the prediction results into a dynamic berth allocation model. This framework accounts for operational constraints, such as tides and water depth adaptability, and is solved using an improved genetic algorithm to obtain efficient solutions. Furthermore, green and low-carbon goals are gradually being integrated into berth allocation research. Liu et al. (2025) [10] constructed a dual-objective model for ship speed optimization and berth allocation and proposed a reinforcement learning-driven non-dominated sorting genetic algorithm. Wang et al. (2022) [11] indirectly optimized berth resource utilization through a dynamic congestion charging mechanism, providing a new perspective for berth allocation optimization.

2.2. Lock Scheduling Problem

Ship lock scheduling, a bottleneck facility in inland waterway and river–sea intermodal transport, aims to minimize ship waiting time and operating costs by optimizing the allocation of lock chambers, navigation sequencing, and in-cabin placement schemes. Lock scheduling problems can be divided into independent and series lock scheduling based on the lock layout. The latter is significantly complex due to the need for coordinating ship transfers between locks.

Early studies on independent lock scheduling primarily focused on accurate modeling. Nauss (2008) [12] constructed an integer programming model for scheduling tugboats and barges in river locks. The aforementioned study demonstrated that the search space for the optimal solution can be significantly reduced under certain conditions, thereby

significantly improving scheduling efficiency compared with the first-come, first-served scheme. Verstichel's team [13–17] carried out a series of pioneering work: in 2009, they proposed a multi-objective optimization model and a delayed acceptance algorithm to balance water consumption and waiting time; in 2011, they decomposed the lock scheduling problem into packing and scheduling subproblems and proposed a metaheuristic algorithm, which greatly improved the solution efficiency; in 2014, they proposed a multi-order optimal adaptation heuristic for the lock chamber placement subproblem, which can quickly solve thousands of instances; in the same year, they constructed an integrated hybrid integer programming model for the generalized lock scheduling problem, which could optimally solve 30 instances within 12 h; in 2015, they effectively shortened the solution time for large-scale instances through combined Benders decomposition, providing key technologies for solving large-scale lock scheduling problems. The scheduling of tandem locks and complex scenarios has been a research focus in recent years. Ji et al. (2021) [18] constructed a hybrid integer programming model for tandem locks from the perspective of flexible workshop scheduling, which can optimally solve the case of multi-compartment locks and multiple ships. In 2022, they [19] further proposed a decomposition framework, combined with tabu adaptive large neighborhood search (ALNS), to address the large-scale problem of generalized tandem lock scheduling. In the field of congestion and green scheduling, Deng et al. (2021) [20] proposed a combination scheme of congestion charging and service time window, which could significantly minimize social costs. Zhao et al. (2022) [21] constructed a dual-objective model and ship reservation system optimization scheme for the congestion problem of the Three Gorges Dam, balancing waiting time and carbon emissions. Wu et al. (2024) [22] constructed a green lock scheduling model, which achieved the coordinated optimization of fuel consumption and total dwell time through large neighborhood search. Ji et al. (2019) [23] focused on the coordinated scheduling of locks and wharves in inland waterways and proposed a hybrid heuristic algorithm driven by fuzzy logic control, which efficiently performed in the data verification of the Three Gorges Dam. Ji et al. (2025) [24] integrated lock scheduling with port resource allocation by developing a mixed-integer programming model and a mixed heuristic framework, providing a new direction for the optimization of port–locks on inland waterways.

2.3. Joint Optimization Problem of Port Berths and Locks

Independent optimization of berths and locks is difficult to adapt to the continuous operation process of ships from locks to ports, often resulting in ship congestion at the connection point. At present, research on the coordinated scheduling of the primary resources of the two remains relatively scarce. Existing related research focuses more on the transfer decisions of locks and ports, rather than the systematic coordinated scheduling of key resources, such as locks, berths, and quay cranes. Consequently, the connection efficiency problem of port–locks is challenging to fundamentally address.

Early related research mostly focused on local connection optimization and did not achieve true resource coordination. Zhang et al. (2016) [25] developed a model to minimize the total waiting time for the channel–berth connection problem of unidirectional channel ports and proposed a simulated annealing–multi-population genetic algorithm. However, the model mainly focused on the scheduling coordination of the channel and berth, without involving the coordinated optimization of lock resources. Ji Bin (2018) [2] focused on the coordination of locks and berths in water conservancy hubs, establishing a mixed-integer programming model that considered the willingness of ships to cross the dam. The model focused on optimizing the decision making of ships passing through the lock and crossing the dam. However, the resource linkage mechanism between berth operations and lock scheduling was relatively limited, and the deep integration of the two operation processes was not achieved. Although some recent studies have addressed the

multi-scenario optimization of port–locks, they have yet to overcome the core bottleneck of resource coordination. Zhang et al. (2023) [1] constructed a multi-objective model for lock utilization, waiting time, and energy consumption and verified its practicality. However, they did not incorporate berth resources into the coordination framework, focusing only on the scheduling optimization of locks and approach channels. Hao et al. (2023) [26] considered the influence of tides and studied the joint scheduling of barges and tugboats in river–sea intermodal transport, focusing on the coordination of transport vehicles rather than the integration of resources within the port–lock. Liu et al. (2024) [3] focused on the coordination of waterways and locks during the flood season, optimizing vessel waiting time. Nonetheless, they did not address the timing connection and resource adaptation between berth operations and lock scheduling.

At present, research on the joint optimization of berths and locks remains relatively scarce. Ji et al. (2025) [24] first explicitly proposed the problem of the coordinated scheduling of port–lock resources, filling this research gap. This study maps the joint scheduling problem into a variant of flexible workshop scheduling, constructing a mixed-integer programming model that integrates lock scheduling, berth allocation, and quay crane allocation. Moreover, this study innovatively combines ALNS, the Floyd–Warshall algorithm, and a multi-order optimal adaptation heuristic to achieve deep collaboration among multiple resources in the port–lock system. Validation based on measured data from the Albert Canal in Belgium and the Three Gorges Dam on the Yangtze River demonstrates that this method can effectively minimize vessel waiting time, clarify the optimal vessel throughput ratio and the number of quay cranes under different traffic volumes, and provide core theoretical support for the efficient operation of the port–lock system. However, they mainly focus on the joint optimization of inland river port–locks, without needing to consider the coordination between seagoing vessels and river vessels.

Independent optimization research on berth allocation and lock scheduling has formed a relatively mature theoretical system and technical methods. Scholars have conducted extensive research on discrete and continuous berth allocation strategies and time-series scheduling rules for locks, proposing various effective solution schemes, such as heuristic algorithms, machine learning models, and mathematical programming. These studies have provided solid support for the optimal allocation of resources in a single link. However, joint optimization research on the synergistic interaction between the two remains relatively scarce. Existing results often neglect the spatiotemporal coupling relationship between berths and locks, especially the spatiotemporal coupling relationship among riverside berths, seaside berths, and locks, making it difficult to adapt to the complete operational process of “ship arrival lock passage–berthing” in port operations. This situation results in insufficient overall and coordinated resource allocation, restricting further improvement in port operation efficiency. Therefore, joint optimization research on berths and locks and the development of an intelligent decision-making framework that takes into account the constraint coupling and dynamic adaptability of the two are of great theoretical and practical significance for overcoming the limitations of single-link optimization and maximizing the overall resource utilization efficiency of ports. The characteristics of the relevant literature are shown in Table 1.

Table 1. Characteristics of lock and berth in the relevant literature.

Literature	Study Object	Direct Transshipment Mode	2D Bin Packing Constraint	Seagoing and River Vessel Coordination
Nauss (2008) [12]	Lock	×	×	×
Verstichel & Berghe (2009) [13]	Lock	×	×	×
Verstichel et al. (2011) [14]	Lock	×	×	×
Verstichel et al. (2014a) [15]	Lock	×	√	×

Verstichel et al. (2014b) [16]	Lock	×	√	×
Verstichel et al. (2015) [17]	Lock	×	√	×
Deng et al. (2021) [20]	Lock	×	×	×
Ji et al. (2021) [18]	Lock	×	√	×
Ji et al. (2022) [19]	Lock	×	√	×
Wang T. & Li (2022) [11]	Lock	×	×	×
Zhao et al. (2022) [21]	Lock	×	×	×
Zhang Y. et al. (2023) [1]	Lock	×	×	×
Liu S et al. (2024) [3]	Lock	×	×	×
Wu et al. (2024) [22]	Lock	×	√	×
Cordeau et al. (2005) [4]	Berth	×	×	×
Zhang X. et al. (2016) [25]	Berth	×	×	×
Agra & Oliveira (2018) [5]	Berth	×	×	×
Correcher et al. (2019) [6]	Berth	×	×	×
Corry & Bierwirth (2019) [7]	Berth	×	×	×
Jia et al. (2020) [8]	Berth	×	×	×
Korekane et al. (2024) [9]	Berth	×	×	×
Li et al. (2024) [27]	Berth	×	×	×
Liu B. et al. (2025) [10]	Berth	×	×	×
Wang L. et al. (2025)	Berth	×	×	×
Zhang Y. et al. (2023) [1]	Berth	×	×	×
Hao et al. (2023) [26]	Berth	×	×	×
Ji (2018) [2]	Ship locks and river port berths	×	√	×
Ji et al. (2019) [23]	Ship locks and river port berths	×	√	×
Ji et al. (2025) [24]	Ship lock, river port berth and quay crane	×	√	×
Our work	Ship lock, seaside and riverside berth	√	√	√

3. Problem Description and Mathematical Model

3.1. Problem Description

This study focuses on the collaborative scheduling problem of a river–sea intermodal transport system under a single port–lock environment. The analysis considers three key facilities: seaside berths, riverside berths, and locks (Figure 1). Unlike the traditional independent operation mode of berth scheduling and lock scheduling, this study focuses on the spatiotemporal coupling relationship between berth operations and lock passage to achieve the overall coordination and integrated optimization of “berth–lock” resources within a port. The system involved in this study includes lock facilities, seaside operating berths, riverside operating berths, and the cargo transport system connecting the river–sea operation areas. The ship-related parameters include the seagoing vessel arrival time, river vessel lock arrival time, vessel dimensions, and load information; the operational parameters include vessel loading and unloading requirements; and facility parameters include lock chamber capacity and berth operation capacity, all of which are considered known. In terms of operational procedures, seagoing vessels berth at seaside berths to complete loading and unloading operations, while river vessels berth at riverside berths after passing through the locks and directly transfer cargo within the port with their corresponding seagoing vessels. Thereafter, these vessels pass through the locks again to transfer cargo to inland areas.

Under the above-mentioned operational procedures, berth scheduling, a fundamental operational link in the river–sea intermodal transport system, mainly addresses specific issues, such as the allocation of seaside and riverside berths, the timing of vessel

berthing operations, and the arrangement of port entry and exit sequences. The decision-making content of this framework includes allocating the corresponding seaside and riverside berths to the seagoing and river vessels, and determining the start and end times of operations and the port entry and exit sequences for various types of vessels. This link must satisfy the following constraints: each vessel occupies only one berth throughout its operational cycle, and berth sharing is not permitted. In this study, direct transfer operations are implemented between river vessels and their corresponding seagoing vessels. The start time of the vessel's operations must not be earlier than its arrival time. The berth operation time interval of seagoing vessels must cover the operation time window of the corresponding river vessels to ensure the continuity of cargo handover. Moreover, the adjacent vessels at the same berth must operate sequentially to avoid operational conflicts. Closely linked to berth scheduling is lock scheduling, a crucial hub connecting river–sea intermodal transport. This mechanism primarily addresses specific issues, such as lock allocation for river vessels, lock chamber allocation, lock passage sequences, and lock opening times. The decision-making process of this mechanism includes assigning each river vessel a unique upstream or downstream lock session, allocating a corresponding lock chamber for each session, and determining the lock opening time and the order in which river vessels enter the locks. The corresponding constraints include the following: each river vessel can only belong to one lock session and is not allowed to pass through multiple lock sessions; the lock opening time of a session cannot be earlier than the arrival time of all river vessels within that session; each lock session corresponds to only one lock chamber; the dimensions of the river vessel must satisfy the capacity requirements and safe passage standards of the allocated lock chamber; and the opening times of adjacent lock sessions in the same or opposite directions must maintain a reasonable interval to avoid passage conflicts.

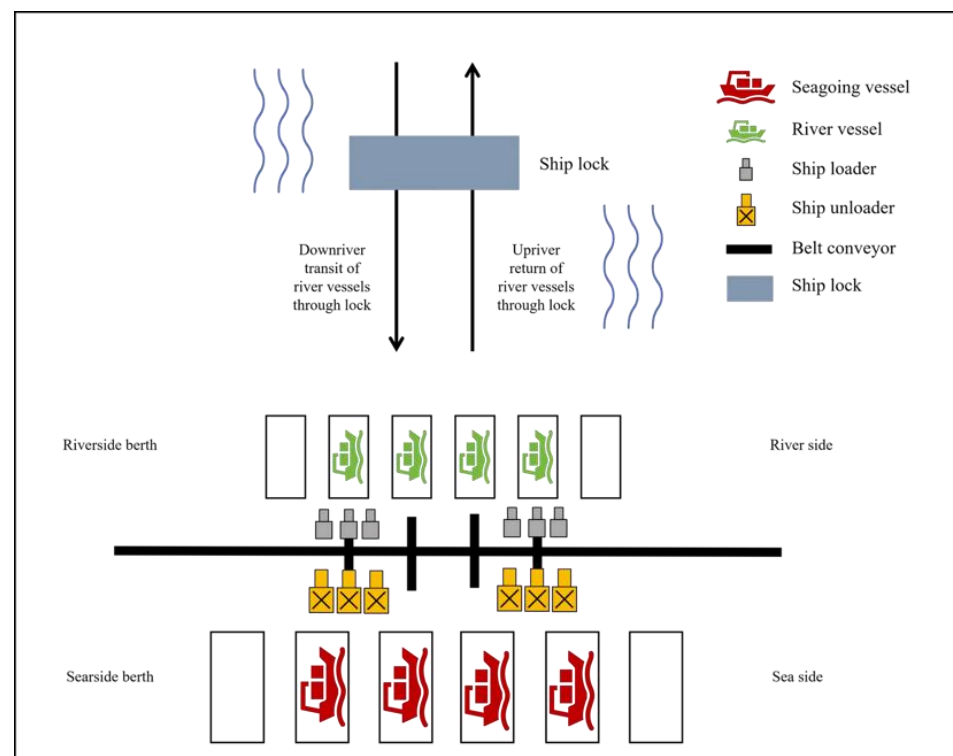


Figure 1. Schematic of port–lock joint scheduling under the river–sea direct transport mode.

Based on the decision-making content and constraints of berth and lock scheduling, the key to the joint scheduling of the river–sea intermodal transport system lies in breaking the boundary of their independent operation, coordinating and integrating the deci-

sion-making elements of the two types of scheduling, and realizing the integrated and coordinated allocation of seaside berths, riverside berths, and lock resources. The primary objective is to coordinate the allocation of berths, berthing sequence, and lock passage arrangements of ships within the system while simultaneously meeting the constraints of berth operations and lock passage. Accordingly, the waste of time and space resources and operational process conflicts that may occur under the traditional independent scheduling mode are prevented, thereby realizing the orderly connection of each operational link. To this end, this study takes improving the overall operational efficiency and service capacity of port river–sea intermodal transport as the guide and minimizing the total system time cost as the optimization objective. The time cost includes the total time of seagoing and river-going vessels in port (including waiting, operation, and port turnaround times) and the time of river-going vessels in locks (including lock waiting and passage times), thereby improving the comprehensive utilization efficiency of berth and lock resources.

Berth scheduling and lock scheduling must be jointly optimized because they are not independent operational links but instead form a closely coupled relationship of mutual constraints and influences in terms of time, space, and resource capacity. The decision-making results of any link will directly affect the operational efficiency of the other link and ultimately influence the operational performance of the entire intermodal transport system. In terms of time, the berth operation time window constitutes an important constraint on lock scheduling: the start and end times of seagoing vessel operations determine the corresponding river vessel operation time range; the completion time of river vessel operations further affects the lock passage demand, and the lock opening time must simultaneously match the river vessel arrival time and the berth operation completion time; if the connection between the two is unbalanced, then river vessels can be left waiting for locks or seagoing vessels can experience delays. Moreover, the lock interval and passage capacity constraints also restrict the berth operation sequence of river vessels, resulting in idle berth or lock resources. Furthermore, the location of the berths on the riverside determines the distance and actual arrival time of river vessels at the locks, thus affecting the lock sequence arrangement. Meanwhile, the lock chamber capacity determines the maximum throughput of a single lock cycle. If these two factors do not match the operational output of the riverside berths, then backlogs of river vessels can easily form after their operations are completed, or berth occupancy can become ineffective. In terms of resource capacity, the overall operational capacity of the berths determines the scale of vessels that can be concurrently served and the intensity of lock throughput demand. The lock's throughput capacity directly affects the efficiency of cargo transshipment for river vessels. If any link becomes a capacity bottleneck, then it will restrict the overall operational capacity of the system, thereby weakening the comprehensive efficiency of the river–sea intermodal transport system.

3.2. Model-Related Assumptions

In light of the complex situation of the joint optimization model of river–sea intermodal transport and lock scheduling, this study makes the following assumptions based on the actual situation:

1. The matching plan of sea ships and river ships in river–sea intermodal transport, the arrival time of sea ships at the port, and the arrival time of river ships at the lock are determined.
2. The berths are all discrete berths, and the draft of the berths satisfies the berthing requirements of the ships.
3. Each berth can only serve one ship at a time.
4. Given that no waterway congestion occurs, the time required for berthing and departure is not considered. This study assumed that the berthing time of the ship is

the start time of loading and unloading. Meanwhile, the departure time of the ship is the end time of loading and unloading.

5. Loading and unloading and horizontal transport are represented by the loading and unloading transport efficiency matrix between different berths.

3.3. Related Parameters and Variables

The parameters and variables in the model are defined as follows based on the above-mentioned model assumptions to further characterize the joint optimization model of river–sea intermodal transport and lock scheduling:

1. Set

C is the set of physical lock chambers, $u \in C$, $u = (1, 2, \dots, Num_c)$, where Num_c is the total number of lock chambers;

DL represents the set of downstream lock batches, $DL = (1, 2, \dots, |\bar{I}_J|)$, where $|\bar{I}_J|$ is the total number of river vessels;

G represents the set of correspondences between seagoing and river vessels, $(i, j) \in G$;

I_H represents the set of seagoing vessels, and $|\bar{I}_H|$ is the total number of all arriving seagoing vessels;

$I_{H'}$ represents the union of the virtual seagoing vessels $\{0\}$ and the set of arriving seagoing vessels I_H ;

I_J represents the set of river vessels, and $|\bar{I}_J|$ is the total number of all arriving river vessels;

$I_{J'}$ represents the virtual river vessels $\{0\}$ and the set of arriving river vessels I_J ;

K_S represents the set of seaside berths, and $|\bar{K}_S|$ is the total number of all seaside berths;

K_R represents the set of riverside berths, and $|\bar{K}_R|$ is the total number of all riverside berths;

Lk represents the set of lock batches, $Lk = (1, 2, \dots, Num)$;

UL represents the set of upstream lock batches, $k \in UL$, $UL = (|\bar{I}_J| + 1, \dots, Num_l)$, where Num_l is the total number of lock cycles, $Num_l = 2 * |\bar{I}_J|$.

2. Related parameters

α indicates the port time weight of seagoing vessels;

β indicates the full process time weight of river vessels;

γ indicates the lock passage time weight of river vessels;

$T_i^{arr} Q_j v_{k,l}$ indicates the transfer efficiency between berths, $k \in K_S, l \in K_R$ (unit: tons/h);

$\max TM Num_c c_d$ is the time cost of a single downstream lock passage;

c_u is the time cost of a single upstream lock passage;

CW is the lock chamber u width, $u \in C$;

CL is the lock chamber u length, $u \in C$;

M indicates a large constant;

$\max T$ is the maximum transport speed between berths;

$\min t$ is the minimum transport speed between berths;

Num_c is the number of lock chambers;

Q_j indicates the load capacity of the river vessel, $j \in I_J$ (unit: tons);

SW_j indicates the width of river vessels, $j \in I_J$;

SL_j indicates the length of river vessels, $j \in I_J$;

T_i^{arr} is the arrival time of vessel i , defined as the port arrival time for seagoing vessels ($i \in I_H$) and the lock arrival time for river vessels ($i \in I_J$);

t_c is the lock chamber processing time;

t_{same} is the time for continuous batch switching in the same direction;

t_{travel} is the time for the river vessel to sail from the riverside berth to the upstream and downstream locks;

$v_{k,l}$ indicates the transfer efficiency between berths, $k \in K_S$, $l \in K_R$ (unit: tons/h).

3. Decision variables

$A_{2,j}$ is a continuous variable, representing the arrival time of the river vessel at the port, $j \in I_J$;

$A_{lock,j}$ is a continuous variable, representing the arrival time of the river vessel at the upstream lock, $j \in I_J$;

$below_{j,j',k}$ is a 0–1 variable, representing the relative position of the vessel (down); $j, j' \in I_J, k \in Lk$;

$c_{down,j}$ is a continuous variable, representing the start time of the river vessel entering the lock downstream; $j \in I_J$;

C_k is a continuous variable, representing the lock completion time; $k \in Lk$;

$c_{up,j}$ is a continuous variable, representing the start time of the river vessel entering the lock on the upstream side, $j \in I_J$;

$f_{down,j}$ is a continuous variable, representing the completion time of the river vessel entering the lock on the downstream side, $j \in I_J$;

$f_{j,k}$ is a 0–1 variable, which is 1 if the river vessel j is assigned to the lock k ; otherwise, 0; $j \in I_J, k \in Lk$;

$f_{up,j}$ is a continuous variable, representing the completion time of the river vessel entering the lock on the upstream side, $j \in I_J$;

$left_{j,j',k}$ is a 0–1 variable, representing the relative position of the vessel (left); $j, j' \in I_J, k \in Lk$;

$proc_{k,u}$ is a 0–1 variable, lock number k 1 if assigned to lock chamber u ; otherwise, 0; $k \in Lk, u \in C$;

$seq_{i,i'}^H$ is a 0–1 variable, 1 if seagoing vessel i is operating at the same berth before seagoing vessel i' ; otherwise, 0. $i, i' \in I_{H'}$;

$seq_{j,j'}^J$ is a 0–1 variable, 1 if river vessel j operates at the same berth before river vessel j' ; otherwise, 0; $j, j' \in I_J$;

$seq_{k,l}^{lock}$ is a 0–1 variable, 1 if lock sequence k is before lock sequence l ; otherwise, 0; $k, l \in Lk$;

$s_{k,l}$ is a 0–1 variable, representing the lock sequence conversion time; if k and l are in the same direction, it is the time for one lock opening; if k and l are in different directions, it is 0; $k, l \in Lk$;

$T_i^{start/end}$ is a continuous variable, representing the start/end times of loading or unloading in vessel i ;

$x_{i,k}$ is a 0–1 variable, 1 if seagoing vessel i is assigned to berth k ; otherwise, 0; $i \in I_{H'}, k \in K_S$;

$x_{pos_{j,k}}$ is a continuous variable, representing the ship's x -coordinate, $j \in I_J, k \in Lk$;

$y_{pos_{j,k}}$ is a continuous variable, representing the ship's y -coordinate, $j \in I_J, k \in Lk$;

$y_{j,l}$ is a 0–1 variable, 1 if river vessel j is assigned to berth l ; otherwise, 0; $j \in I_J, l \in K_R$;

$z_{i,k,j,l}$ is a 0–1 variable, 1 if seagoing vessel i is at berth k , and its corresponding river vessel j is at berth l ; $i \in I_{H'}, j \in I_J, k \in K_S, l \in K_R$;

z_k is a 0–1 variable, 1 if the lock is used; otherwise, 0; $k \in Lk$.

3.4. Model Construction

3.4.1. Objective Function

The objective of this model is to minimize the total time cost of the system. Specifically, the goal of the joint optimization of river–sea intermodal transport and lock scheduling is to reduce the weighted sum of the port time cost of seagoing and river vessels and the lock time cost of river vessels, covering the three core dimensions of the time cost of seagoing vessels in port, the full-process time cost of river vessels, and the lock operation cost.

$$\min Z = \alpha \cdot \sum_{i \in I_H} (T_i^{\text{end}} - T_i^{\text{arr}}) + \beta \cdot \sum_{j \in I_J} ((c_{\text{down},j} - T_j^{\text{arr}}) + (T_j^{\text{end}} - A_{2,j}) + (c_{\text{up},j} - A_{\text{lock},j})) + \gamma \cdot (c_u \cdot \sum_{k \in UL} z_k + c_d \cdot \sum_{k \in DL} z_k) \quad (1)$$

3.4.2. Constraints

The joint optimization of river–sea intermodal transport and lock scheduling follows a series of orderly processes. Close coordination is required between various vessels and equipment to ensure the rapid and efficient transfer of dry bulk cargo. The main constraints include berth allocation constraints, sea and river vessel operation time coordination constraints, river vessel lock passage time constraints, and lock bidirectional scheduling constraints. The specific constraints are as follows.

1. Berth allocation constraints

$$\sum_{k \in K_S} x_{i,k} = 1 \quad \forall i \in I_H; \quad (2)$$

$$\sum_{l \in K_R} y_{j,l} = 1 \quad \forall j \in I_J; \quad (3)$$

$$z_{i,k,j,l} \leq x_{i,k} \quad \forall (i,j) \in G, k \in K_S, l \in K_R; \quad (4)$$

$$z_{i,k,j,l} \leq y_{j,l} \quad \forall (i,j) \in G, k \in K_S, l \in K_R; \quad (5)$$

$$z_{i,k,j,l} \geq x_{i,k} + y_{j,l} - 1, \quad \forall (i,j) \in G, k \in K_S, l \in K_R. \quad (6)$$

Constraints (2) and (3) represent the uniqueness of berth allocation for sea and river vessels, respectively; constraints (4) and (5) represent the matching relationship between sea and river vessels, respectively; and constraint (6) is a mandatory association constraint for pairing sea and river vessels.

2. Sea and river vessel operation time coordination constraints

$$T_i^{\text{start}} \geq T_i^{\text{arr}}, \quad \forall i \in I_H; \quad (7)$$

$$T_j^{\text{start}} \geq A_{2,j}, \quad \forall j \in I_J; \quad (8)$$

$$T_i^{\text{start}} \leq T_j^{\text{start}} \quad \forall (i,j) \in G, i \in I_H; \quad (9)$$

$$T_j^{\text{end}} \leq T_i^{\text{end}} + M \cdot (1 - z_{i,k,j,l}) \quad \forall (i,j) \in G, k \in K_S, l \in K_R; \quad (10)$$

$$T_j^{\text{end}} \geq T_j^{\text{start}} + \frac{Q_j}{v_{k,l}} \cdot z_{i,k,j,l} \quad \forall i \in I_H, \forall j \in I_J; \quad (11)$$

$$T_{i'}^{\text{start}} - T_i^{\text{end}} + M \cdot (1 - seq_{i,i'}^H) \geq 0, \quad \forall i \in I_H, i' \in I_H, i \neq i', k \in K_S; \quad (12)$$

$$T_{j'}^{\text{start}} - T_j^{\text{end}} + M \cdot (1 - seq_{j,j'}^J) \geq 0, \quad \forall j \in I_J, j' \in I_J, j \neq j', l \in K_R; \quad (13)$$

$$\sum_{i \in I_{H'}} (seq_{0,i}^H + seq_{|I_H|+1,i}^H) = |\overline{K_S}|; \quad (14)$$

$$\sum_{i \in I_{H'}} (seq_{i,0}^H + seq_{i,|I_H|+1}^H) = |\overline{K_S}|; \quad (15)$$

$$\sum_{j \in I_{J'}} (seq_{0,j}^J + seq_{|I_J|+1,j}^J) = |\overline{K_R}|; \quad (16)$$

$$\sum_{j \in I_J} (seq_{j,0}^J + seq_{j,|I_J|+1}^J) = |\overline{K_R}|; \quad (17)$$

$$seq_{0,0}^H = 0; \quad (18)$$

$$seq_{0,0}^J = 0; \quad (19)$$

$$\sum_{j \in I_H: j \neq i} seq_{i,j}^H = 1 \quad \forall i \in I_H; \quad (20)$$

$$\sum_{j \in I_H: j \neq i} seq_{j,i}^H = 1 \quad \forall i \in I_H; \quad (21)$$

$$\sum_{j \in I_H: j \neq i} seq_{i,j}^H = 1 \quad \forall i \in I_H; \quad (22)$$

$$\sum_{j \in I_J: j \neq i} seq_{j,i}^J = 1 \quad \forall i \in I_J; \quad (23)$$

$$x_{i,k} + x_{i',k} - M \cdot (2 - seq_{0,i}^H - seq_{|I_H|+1,i}^H - seq_{0,i'}^H - seq_{|I_H|+1,i'}^H) \leq 1 \quad \forall i, i' \in I_H, i \neq i', k \in K_S \quad (24)$$

$$x_{i,k} - x_{i',k} + M \cdot (1 - seq_{i,i'}^H) \geq 0 \quad \forall i, i' \in I_H, i \neq i', k \in K_S; \quad (25)$$

$$x_{i,k} - x_{i',k} - M \cdot (1 - seq_{i,i'}^H) \leq 0 \quad \forall i, i' \in I_H, i \neq i', k \in K_S; \quad (26)$$

$$y_{j,k} + y_{j',k} - M \cdot (2 - seq_{0,j}^J - seq_{|I_J|+1,j}^J - seq_{0,j'}^J - seq_{|I_J|+1,j'}^J) \leq 1 \quad \forall j, j' \in I_J, j \neq j', k \in K_R; \quad (27)$$

$$y_{j,k} - y_{j',k} + M \cdot (1 - seq_{j,j'}^J) \geq 0 \quad \forall j, j' \in I_J, j \neq j', k \in K_R; \quad (28)$$

$$y_{j,k} - y_{j',k} - M \cdot (1 - seq_{j,j'}^J) \leq 0, \forall j, j' \in I_J, j \neq j', k \in K_R; \quad (29)$$

$$seq_{i,i'}^H + seq_{i',i}^H \leq 1 \quad \forall i, i' \in I_H: i \neq i'; \quad (30)$$

$$seq_{j,j'}^J + seq_{j',j}^J \leq 1 \quad \forall j, j' \in I_J: j \neq j'. \quad (31)$$

Constraints (7) and (8) represent the start time of a seagoing and river vessel, respectively; constraint (9) represents the synchronous start of a seagoing and river vessel; constraint (10) represents the end time of river vessels no later than that of seagoing vessels; constraint (11) represents the lower limit of a river vessel's operation time; constraints (12) and (13) represent the operation sequence constraints of a seagoing and river vessel in the same berth, respectively; constraints (14)–(29) collectively model the berth service sequences as a set of ordered chains. Specifically, constraints (14)–(19) define the boundary conditions by anchoring each chain to virtual source and sink nodes, with the number of chains equal to the number of available berths. Constraints (20)–(23) impose the uniqueness of each vessel's predecessor and successor within its chain. Constraints (24) and (27) prevent two vessels from being simultaneously assigned to the same berth unless one of them occupies a chain boundary position. Constraints (25), (26), (28), and (29) enforce consistency between the sequencing variables and the berth assignment variables, ensuring that consecutively sequenced vessels are always assigned to the same berth.

3. Lock allocation and constraints

$$f_{j,k} \leq z_k, \forall j \in I_j, k \in Lk; \quad (32)$$

$$\sum_{u \in Cs} proc_{k,u} = z_k, \forall k \in Lk; \quad (33)$$

$$proc_{k,u} + \sum_{v \in C, v \neq u} proc_{l,v} + seq_{k,l}^{lock} \leq 2, \forall k, l \in Lk, k < l, u \in C; \quad (34)$$

$$C_l - C_k + M * (3 - seq_{k,l}^{lock} - proc_{k,u} - proc_{l,u}) \geq t_c + s_{k,l} \cdot t_{same}, \forall k, l \in Lk, k < l, u \in C \quad (35)$$

$$C_k - C_l + M * (2 + seq_{k,l}^{lock} - proc_{k,u} - proc_{l,u}) \geq t_c + s_{k,l} \cdot t_{same}, \forall k, l \in Lk, k < l, u \in C \quad (36)$$

Constraint (32) represents the association between lock usage and river vessel allocation; constraint (33) represents the uniqueness of lock usage and lock chamber allocation; and constraints (34)–(36) represent the time constraints when lock usage is allocated to the same lock chamber.

Downstream lock:

$$\sum_{k \in DL} f_{j,k} = 1, \forall j \in I_j; \quad (37)$$

$$z_{k+1} \leq z_k, \forall j \in I_j, k \in DL, k < |\bar{I}_j|; \quad (38)$$

$$c_{down,j} \leq (C_k - t_c) + M \cdot (1 - f_{j,k}), \forall j \in I_j, k \in DL; \quad (39)$$

$$c_{down,j} \geq (C_k - t_c) - M \cdot (1 - f_{j,k}), \forall j \in I_j, k \in DL. \quad (40)$$

Constraint (37) indicates the uniqueness of the downstream lock allocation for river vessels; constraint (38) indicates the constraint of the continuous use of downstream lock sessions; and constraints (39) and (40) indicate the correlation between the downstream lock entry time and the lock session completion time for river vessels.

Upward lock:

$$\sum_{k \in UL} f_{j,k} = 1, \forall j \in I_j; \quad (41)$$

$$z_{k+1} \leq z_k, \forall j \in I_j, k \in UL, k < Num_l; \quad (42)$$

$$c_{up,j} \leq (C_k - t_c) + M \cdot (1 - f_{j,k}), \forall j \in I_j, k \in UL; \quad (43)$$

$$c_{up,j} \geq (C_k - t_c) - M \cdot (1 - f_{j,k}), \forall j \in I_j, k \in UL. \quad (44)$$

Constraint (41) indicates the uniqueness of the upstream lock allocation for river vessels; constraint (42) indicates the constraint of the continuous use of the upstream lock; and constraints (43) and (44) indicate the correlation between the upstream entry time of river vessels and the lock completion time.

Packing and size constraints:

$$x_{pos,j,k} + SW_j \leq CW + M \cdot (1 - f_{j,k}), \forall j \in I_j, k \in Lk; \quad (45)$$

$$y_{pos,j,k} + SL_j \leq CL + M \cdot (1 - f_{j,k}), \forall j \in I_j, k \in Lk; \quad (46)$$

$$x_{pos,j,k} \geq 0, \forall j \in I_j, k \in Lk; \quad (47)$$

$$y_{pos,j,k} \geq 0, \forall j \in I_j, k \in Lk; \quad (48)$$

$$left_{j,j',k} + left_{j',j,k} + below_{j,j',k} + below_{j',j,k} \geq 1, \forall j, j' \in I_j, j < j', k \in Lk; \quad (49)$$

$$x_{pos_{j,k}} + SW_j \leq x_{pos_{j',k}} + M \cdot (1 - left_{j,j',k}) + M \cdot (2 - f_{j,k} - f_{j',k}) \\ \forall j, j' \in I_j, j < j', k \in Lk; \quad (50)$$

$$y_{pos_{j,k}} + SL_j \leq y_{pos_{j',k}} + M \cdot (1 - below_{j,j',k}) + M \cdot (2 - f_{j,k} - f_{j',k}) \\ \forall j, j' \in I_j, j < j', k \in Lk; \quad (51)$$

$$x_{pos_{j,k}} + SW_j + MW \leq x_{pos_{j',k}} + M \cdot (1 - left_{j,j',k}) + \\ M \cdot (2 - f_{j,k} - f_{j',k}), \forall j, j' \in I_j, j \neq j', k \in Lk; \quad (52)$$

$$y_{pos_{j,k}} + SL_j + ML \leq y_{pos_{j',k}} + M \cdot (1 - below_{j,j',k}) + \\ M \cdot (2 - f_{j,k} - f_{j',k}), \forall j, j' \in I_j, j \neq j', k \in Lk; \quad (53)$$

$$x_{pos_{j,k}} \leq M \cdot f_{j,k}, \forall j \in I_j, k \in Lk; \quad (54)$$

$$y_{pos_{j,k}} \leq M \cdot f_{j,k}, \forall j \in I_j, k \in Lk; \quad (55)$$

$$left_{j,j',k} \leq f_{j,k}, \forall j, j' \in I_j, j < j', k \in Lk; \quad (56)$$

$$left_{j,j',k} \leq f_{j',k}, \forall j, j' \in I_j, j < j', k \in Lk; \quad (57)$$

$$below_{j,j',k} \leq f_{j,k}, \forall j, j' \in I_j, j < j', k \in Lk; \quad (58)$$

$$below_{j,j',k} \leq f_{j',k}, \forall j, j' \in I_j, j < j', k \in Lk. \quad (59)$$

Constraints (45)–(48) indicate the size adaptation constraint between river vessels and lock chambers, and constraints (49)–(59) indicate the relative position and safety distance constraint of river vessels within the lock.

4. Time connection constraints between berth allocation and lock scheduling:

$$A_{2,j} = f_{down,j} + t_{travel}, \forall j \in I_j; \quad (60)$$

$$A_{lock,j} = T_j^{end} + t_{travel}, \forall j \in I_j; \quad (61)$$

$$f_{down,j} = c_{down,j} + t_c, \forall j \in I_j; \quad (62)$$

$$f_{up,j} = c_{up,j} + t_c, \forall j \in I_j; \quad (63)$$

$$c_{up,j} \geq A_{lock,j}, \forall j \in I_j; \quad (64)$$

$$c_{down,j} \geq T_j^{arr}, \forall j \in I_j. \quad (65)$$

Constraints (60)–(65) indicate the time connection constraints of the entire process of river vessels.

4. Port–Lock Collaborative Optimization Algorithm Design

The primary objective of the port–lock collaborative scheduling algorithm is to collaboratively optimize the berth allocation and lock scheduling of seagoing and river vessels to minimize the total time of vessels in port. A close relationship exists between lock and berth scheduling. On the one hand, the completion time of a vessel's operations at the berth directly affects its arrival time at the lock, thereby influencing the composition and scheduling order of lock batches. On the other hand, the lock's passage time and batch arrangement constrain the vessel's operation time window within the port, thereby affecting the feasibility and efficiency of berth allocation. Separate optimization of these two aspects often results in local optima and may even cause resource conflicts and time delays. Therefore, this study adopts a joint optimization approach and designs a collaborative optimization iterative algorithm for locks and berths.

The algorithm adopts an iterative algorithm framework, gradually approximating the global optimal solution through the combination of lock and berth allocation. This framework draws on the hierarchical iterative idea for solving complex systems. For example, in dealing with certain problems, such as cooperative control of connected autonomous vehicles, dynamic scheduling of AGVs in automated terminals, and berth planning under uncertain environments, research has shown that decomposing the original problem into sub-modules at different levels and designing an iterative cooperative mechanism can effectively minimize the complexity of the problem and improve the overall performance of the system. Therefore, this study adopts an iterative algorithmic framework to address the cooperative optimization problem between port berth and lock allocation and designs heuristic and ALNS algorithms to solve the resulting subproblems based on their structural characteristics.

The collaborative optimization algorithm proposed in this study adopts a hierarchical iterative structure, and the overall process is shown in Figure 2. Its computational complexity is $(|\bar{I}_H|^3 + |\bar{I}_J|^3)$.

Step 1: Initialization and data reading

Read ship static data (size, cargo capacity, etc.), berth information, inter-port transfer speed, lock parameters, etc. Record the original descent time of the river ship to the lock, and initialize the global optimal solution.

Step 2: Outer iteration

Set the maximum number of outer iterations T , and initialize the outer iteration counter $t = 1$. Execute the following steps in a loop until $t > T$:

Step 2.1 Downward ship lock dispatch

Two-dimensional packing strategy: Arrange the ship layout based on the lock chamber size and safety clearance constraints.

Ship lock allocation: Determine the passage time and number of locks for each batch of ships.

Output the ship lock scheduling plan.

Step 2.2: Inner layer optimization: Cooperative optimization of ALNS berth allocation and lock scheduling

Execute the following collaborative optimization process based on the current arrival time of the river vessel at the lock:

Step 2.2.1: Berth allocation optimization (ALNS framework)

Initial solution generation: Allocate berths for sea and river vessels based on the order of arrival time.

Destruction stage: Random ship destruction operator, random river ship destruction operator, and random high-cost destruction operator.

Repair phase: Greedy repair operator, regret value repair operator.

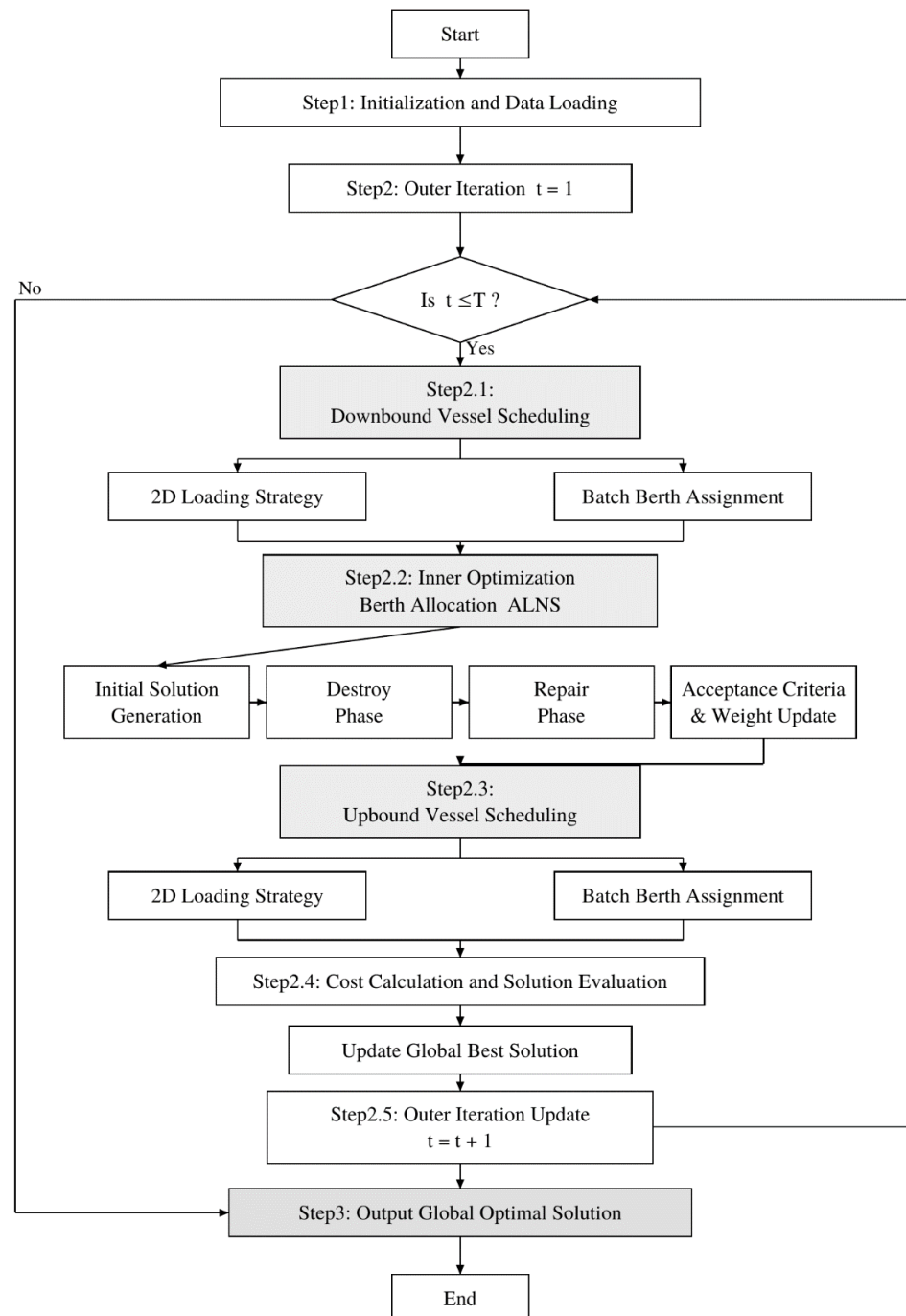


Figure 2. Schematic of port–lock joint scheduling under the river–sea direct transport mode.

Acceptance criteria: The simulated annealing mechanism determines whether to accept the new solution.

Weight update: Regularly update operator weights.

Output the current optimal berth plan and the completion time of the river vessel operation.

Step 2.3: Upstream ship lock optimization (heuristic packing strategy)

Two-dimensional packing strategy: Arrange the ship layout based on the lock chamber size and safety clearance constraints.

Ship lock allocation: Determine the passage time and number of locks for each batch of ships.

Output the ship lock scheduling plan.

Step 2.4: Cost calculation and evaluation of solutions

Calculate the total cost of the current plan, and update the global optimal solution.

Step 2.5: Outer iteration update

$t = t + 1$; Extract the completion time of this round of river vessel operations, and calculate the upstream return time. Accordingly, further optimize the downstream lockage time and batch scheduling.

Step 3: Algorithm termination and output

Terminate when the maximum number of iterations in the outer layer is reached. Output the globally optimal berth allocation plan and ship lock scheduling plan. Otherwise, return to Step 2.1.

4.1. Ship Lock Allocation Algorithm

The preliminary step of the entire iterative algorithm is the design of the ship lock allocation algorithm to define a feasible range of time and space for subsequent berth allocation. This step determines the reasonable number of locks and the layout of the space inside the locks through ship lock allocation. The primary steps include 2D packing and ship lock allocation strategies (Algorithm A1).

4.1.1. 2D Packing Strategy

This study assumes that the width of ship s is SW_s , the length is SL_s , the upper limit of the width of lock chamber c is CW_{cmax} , the upper limit of the length is CL_{cmax} , the lateral safety clearance is δ_w , and the longitudinal safety clearance is δ_l . In candidate batch b , the cumulative width of the currently arranged ships is SW_b , and the cumulative length is SL_b .

When adding ship s , the algorithm first tries to arrange them side-by-side in the lateral direction:

1. Calculate the total width after lateral arrangement $SW_b = SW_b + SW_s + \delta_w$;
2. Calculate the total length after lateral arrangement $SL_b = \max(SL_b, SL_s)$;
3. If $SW_b \leq SW_{cmax}$ and $SL_b \leq SL_{cmax}$, then adopt the transverse arrangement. If the transverse arrangement is not feasible, then try the longitudinal sequential arrangement:
 - ① Calculate the total width after the longitudinal arrangement $SW_b = \max(SW_b, SW_s)$;
 - ② Calculate the total length after the longitudinal arrangement $SL_b = SL_b + SL_s + \delta_l$;
 - ③ If $SW_b \leq SW_{cmax}$ and $SL_b \leq SL_{cmax}$, then adopt the longitudinal arrangement.

4.1.2. Lock Allocation Strategy

The algorithm adopts a heuristic strategy based on arrival time sequence processing, combined with a batch allocation method based on cost assessment. In each ship to be arranged s , the algorithm traverses all created batches and checks whether it can be added to an existing batch.

Calculate the cost increment b for ships s and candidate batches b . The increment consists of two main parts: the total waiting time of existing vessels in batch ΔC_{add} for adding to an existing batch b due to waiting for s to arrive and the waiting time of vessel s in batch b . Moreover, the fixed cost b_{new} is calculated, including the basic cost C_{new} of creating a new batch C_{batch} of the new batch (usually set to one) and the waiting cost of vessel s in the new batch. The algorithm selects the option with the lower cost: if $\Delta C_{add} < C_{new}$, then s is added to batch b ; otherwise, s , a new batch, is created. The algorithm pseudocode is provided by the Appendix A.

4.2. Berth Allocation Algorithm

The berth allocation algorithm based on ALNS has the primary task of planning specific berths and operation time windows for each seagoing vessel and simultaneously allocating suitable river vessel berths for the associated river vessels in the same port. The optimization goal of this algorithm is to minimize the total time of ships in port (including waiting and operation times).

4.2.1. Initial Solution Generation

An initial feasible solution is generated using a rule-based heuristic construction method. The algorithm executes the following steps:

Step 1: Vessel sequence generation. Sort all vessels (including seagoing vessels and river vessels) according to their arrival times at the port to form a processing sequence.

Step 2: Seagoing vessel berth allocation. Assign a seagoing vessel berth to each seagoing vessel in turn, and the conditions include the following:

- It must be a seagoing vessel berth;
- The berth is available when the seagoing vessel arrives;
- The berth is not occupied during the expected operation period.

The seagoing vessel operation time is estimated based on the total cargo volume of all associated river vessels at a fixed transfer rate. If an available berth is found, then allocate it, calculate its start and end times, and update the berth availability time; if no berth is available, then place the seagoing vessel in the waiting queue.

Step 3: River vessel berth allocation. Sequentially assign berths to each river vessel, under the following conditions:

- Its associated seagoing vessel has been assigned a berth, and its start time has been determined;
- An available river vessel berth is found within the same port (the berth is vacant, and no time conflict occurs).

The river vessels' operating time is determined by its cargo volume and the specific transfer rate between berths. If the conditions are met, then the berth is assigned, and the time is updated; if not (a berth is unavailable or a seagoing vessel is not scheduled), the river vessel is placed in the waiting queue.

Step 4: Waiting queue processing

The berth vacancy time requirement is relaxed for vessels in the waiting queues of seagoing vessels and river vessels, allowing vessels to operate after waiting for berths to become vacant. All vessels must obtain berths by retrying the assignment, ultimately forming a complete initial feasible solution.

4.2.2. Design of the Destruction Operator and Its Update Mechanism

1. Random seagoing vessel destruction operator

Generate a random number k_1 in the interval $[1, n_1 \cdot |\bar{I}_H|]$ (where n_1 are the parameters of the random seagoing vessel destruction operator). Randomly select k_1 seagoing vessel tasks from the task list of the current solution. If a seagoing vessel task is selected, then remove the seagoing vessel task and all its subordinate river boat sub-tasks. Moreover, put all removed tasks into the list to be reinserted.

2. Random river boat destruction operator

Generate a random number k_2 in the interval $[1, n_2 \cdot \lceil \bar{I}_j \rceil]$ (where n_2 is the parameter for the random river vessel destruction operator). Randomly select k_2 river vessels from all the assigned river vessels in the current solution. Remove all selected river vessels, change their status to “unassigned”, and add them to the list to be reinserted.

3. High-cost vessel destruction operator

In each vessel (sea and river vessels) in the current solution, calculate its cost contribution value, which is the sum of the vessel’s waiting time (start time minus arrival time) and its operating time. Thereafter, sort all vessels according to this cost contribution value from high to low. In this sorted list, randomly select an integer k_3 between $[1, n_3 \cdot (\lceil \bar{I}_H \rceil + \lceil \bar{I}_j \rceil)]$ (where n_3 is the parameter for the high-cost vessel destruction operator), and remove the top k_3 vessels with the highest cost from the beginning of the list. If the removed vessel is a seagoing vessel, then all of its associated river vessels will also be removed; if the removed vessel is a river vessel, then only the vessel itself will be removed. All removed vessels will be added to the list to be reinserted.

4. Update mechanism of partial solution after disruption

First, identify all affected berths. Thereafter, recalculate the start and end times of each vessel’s operation at the berth according to the original task order, and uniformly update the berth status. Given that the operation times of seagoing vessels and their associated river vessels are interdependent, any adjustment to one side may trigger cascading updates—for example, a change in the start time of a seagoing vessel will affect all associated river vessels, and a change in the overall progress of river vessels may in turn correct the end time of seagoing vessels. The partial solution maintains the feasibility of the remaining scheduling plan while releasing the resource time window through this series of adjustments, thus creating conditions for the reinsertion of subsequent vessels.

4.2.3. Design of the Repair Operator

The repair operator reinserts all vessels removed in the previous step into the partial solution to form a new complete solution. The strategy for repair insertion is to achieve the smallest possible cost increment. Here, two types of insertion operators are used: the greedy repair operator and the regret value insertion operator.

1. Greedy repair operator

First, arrange the ships to be inserted according to their original arrival times. Thereafter, process the ships one-by-one in this order: list all feasible insertion schemes for the ships to be arranged. Each scheme must determine the berth selected by the ship and the start time of operation. Calculate the increase in the objective function value of each feasible insertion scheme. The insertion scheme that minimizes the increase in the objective function value for insertion is inserted. Insert the ships to be arranged in sequence until the list of ships to be inserted is empty.

2. Regret value insertion operator

List all feasible insertion schemes for each ship to be arranged, and calculate the cost of all the insertion schemes. Let the optimal cost be c_1 and the suboptimal cost be c_2 . The regret value of the ship can be expressed as $\Delta c = c_2 - c_1$. Calculate the regret value of each ship to be inserted. Select the ship with the largest regret value Δc , and insert it into the position corresponding to its optimal cost c_1 . Next, given that this insertion changes the availability time of some resources, relist all feasible insertion schemes for the remaining ships to be inserted. Subsequently, calculate the regret value. Thereafter, insert the ship with the highest regret value into the optimal position. The above-mentioned steps are repeated until all ships have been inserted.

4.2.4. Random Perturbation

This study embeds a global optimal solution perturbation mechanism based on iterative stagnation into the adaptive large neighborhood search framework: a deep “destruction–repair” perturbation is implemented when the number of consecutive unimproved iterations reaches the stagnation threshold $\eta = 100$ as the trigger condition: first, seagoing or river vessels are randomly selected as destruction targets with equal probability, and a corresponding number of vessel tasks are removed from the global optimal solution according to a preset ratio (60%); second, feasible berths and operating periods are redistributed to the removed vessels through random insertion heuristics.

4.2.5. Solution Acceptance Rule

This study uses simulated annealing as the acceptance criterion for new solutions. The core of this criterion is to allow the algorithm to accept solutions worse than the current solution with a certain probability. This probability is dynamically adjusted as the search process progresses.

Specifically, let the total cost of the current solution be C_{cur} and the total cost of the newly generated solution be C_{new} . If the new solution is better (i.e., $C_{new} < C_{cur}$), then the new solution is unconditionally accepted, replacing the current solution. If the new solution is worse than the current solution (i.e., $C_{new} \geq C_{cur}$), then it is not directly rejected, but accepted with a certain probability P . The formula for calculating the acceptance probability P is as follows:

$$P = \exp\left(\frac{C_{cur} - C_{new}}{T}\right),$$

where T is the current temperature parameter.

The temperature T is initially set to a relatively high value T_0 (set to 1200 in this study), which makes the algorithm likely to accept inferior solutions in the early stages of the search, thus encouraging extensive exploration of the solution space. The temperature gradually decreases according to the cooling rate α (taken as $\alpha = 0.95$ in this study) as the iteration progresses (i.e., $T \leftarrow \alpha \cdot T$). The decrease in temperature reduces the probability of accepting inferior solutions, and the algorithm’s search behavior gradually shifts from global exploration to a more precise exploration of the neighborhood of the current better solution.

4.2.6. Solution Selection

In the initial stage, all operators have equal weights, and the score is initialized to one. During each iteration, scores are given in a stepwise manner based on the performance of the operator, with higher scores indicating better operator performance. This study sets the following four scenarios: (1) If the new solution is better than the currently known global optimal solution of the algorithm, then a reward score will be given w_1 . If the new solution is better than the current solution but not better than the global optimal solution, then a reward score will be given w_2 . If the new solution is accepted, then reward points will be awarded w_3 . If the new solution is rejected, then a penalty score will be given w_4 . In each segment, record the total reward score obtained after each operator is called σ_i , and the number of times it has been selected n_i . Use exponential smoothing to update the weights of each operator.

$$w_{i+1} = \lambda \cdot w_i + (1 - \lambda) \cdot \frac{\sigma_i}{n_i}$$

In each iteration, select the operator to be used based on the updated weights according to the roulette wheel.

4.3. Algorithm Termination Rules

4.3.1. Inner Iteration Termination Rules

1. Maximum number of iterations: Set a fixed maximum number of iterations $MAX_{interactions}$. When the search process reaches this number, the search stops regardless of the current solution exploration status.
2. Simulated annealing temperature threshold: With the algorithm iteration and the setting of the cooling rate, the cooling rate of simulated annealing is 0.95. When the temperature T drops to an extremely low preset threshold (such as close to zero), the probability of the algorithm accepting a bad solution is negligible, the search process stabilizes, and the inner iteration is terminated at this time.

Stop running and return the optimal berth allocation scheme it has searched when any condition is met.

4.3.2. Termination Condition of the Outer Iteration

1. Maximum number of iterations: Set a maximum number of outer iterations T_{max} . When the number of iterations of the main loop reaches this upper limit, the algorithm terminates. This condition is mainly used to control the overall calculation time.
2. Convergence judgment: At the end of each outer iteration, the algorithm records and compares the historical best total cost C_{best} . If no improvement is obtained in multiple consecutive iterations C_{best} (i.e., the cost has not decreased), the algorithm is judged to have sufficiently converged, and the possibility of continuing to iterate and improve the total cost is low. At this time, the algorithm will also terminate early to conserve computing resources.

When any of the above-mentioned outer termination conditions are met, the entire collaborative optimization algorithm stops running and outputs the historical best solution recorded in the entire process (i.e., the corresponding optimal berth and lock scheduling schemes).

5. Numerical Experiment and Analysis

This section verifies the effectiveness of the proposed port–lock collaborative scheduling model and algorithm through numerical experiments. All experiments were conducted on the same platform: the hardware consisted of an AMD Ryzen 7 4800H CPU with Radeon Graphics and 16 GB of RAM; the software environment was a Windows 10 operating system; and the experiments were implemented using Java in IntelliJ IDEA. As a benchmark, the mathematical programming model constructed in this study was solved using the commercial solver CPLEX 12.8.0, with a maximum solution time of 5 h to obtain the best possible feasible solution within a reasonable timeframe.

5.1. Experimental Data and Case Design

The experimental case design closely follows the typical actual operation scenario of Chinese dry bulk cargo river–sea intermodal transport. The ship parameters were all set according to the standard of the main ship type for coastal dry bulk cargo transport in China and the standard of inland waterway ship type after the opening of the Pinglu Canal. Seagoing vessels cover 12 types of tonnage ranging from 3000 tons to 300,000 tons. Meanwhile, river vessels cover four types of tonnage ranging from 1000 tons to 5000 tons. The detailed parameters are shown in Tables 2 and 3.

Table 2. Seagoing vessel parameters.

No. of Seagoing Vessel	Total Length (m)	Breadth (m)	Depth (m)	Full-Load Draft (m)	Cargo Capacity (10,000 Tons)
H1	96	16.6	7.8	5.8	0.3
H2	115	18.8	9.0	7.0	0.5
H3	135	20.5	11.4	8.5	1.0
H4	164	25.0	13.5	9.8	2.0
H5	190	30.4	15.8	11.2	3.5
H6	223	32.3	17.9	12.8	5.0
H7	228	32.3	19.6	14.2	7.0
H8	250	43.0	20.3	14.5	10.0

Table 3. River vessel parameters.

River Vessel Number	Total Length (m)	Total Width (m)	Full Load Draft (m)	Intermodal Transport Capacity (Tons/Vessel)
J1	45.0~50.0	10.8~11.0	2.6~3.0	0.1
J2	68.0~74.0	13.8~14.0	2.8~3.5	0.2
J3	74.0~80.0	15.5~15.8	3.4~3.8	0.3
J4	86.0~90.0	15.5~15.8	3.2~3.5	0.3
J5	86.0~90.0	15.5~15.8	4.8~5.2	0.5

The port layout adopts a combination of “shore-type seagoing vessel terminal + breakwater-type river vessels terminal”. This experiment focuses on the port’s submerged scenario, and the design parameters for its discrete berths are shown in Table 4. The lock is a single-stage double-line lock. The relevant lock parameters are shown in Table 5.

Table 4. Port berth parameters.

Parameter Type	Value (Tons)
Design Tonnage of Seagoing Vessel Berths (tons)	100,000
Design Tonnage of River Vessel Berths (tons)	5000

Table 5. Lock parameters.

Number of Lock Chambers	Lock Chamber Type	Effective Dimensions (Length × Width × Threshold Depth, m)	Single Operation Time (h)	Same-Direction Re-set Time (h)	Lateral Safety Clearance (m)	Longitudinal Safety Clearance (m)	Port Distance from Lock Travel Time (h)
2	Single-stage, double-line	300 × 34 × 8	1	0.5	2.0	5.0	0.5

Cargo transfer is completed through an “integrated loading/unloading-belt conveyor system”. The transfer efficiency decreases in a stepwise manner with an increase in the distance between berths. Meanwhile, the transfer speed progressively decreases from 5000 to 3000 tons/h, with each step decreasing by 500 tons/h. This speed gradient quantifies the spatial distance difference into operational efficiency parameters, thus reflecting the distance cost of horizontal transportation in the model.

Eight incrementally large test case scenarios were designed to evaluate the algorithm’s performance under different scheduling pressures. In each scale, five different planned arrival times for seagoing vessels were set (8:00–10:00 to 8:00–14:00), and the arrival time for river vessels downstream was set to within 1 h of the seagoing vessel arrival time, simulating operational conditions from relaxed to strained. A total of 40 test instances were generated. The test case scale configuration is shown in Table 6, and the instance naming convention is “total number of seagoing vessel berths @ total number of river vessel berths @ number of seagoing vessels @ number of river vessels @ seagoing vessel arrival time window”. The test cases were divided into four levels—small-scale,

medium-scale, medium-large-scale, and large-scale—primarily based on the number of river vessels for ease of differentiation.

Table 6. Test case scale design.

Instance Category	Case Naming	Seagoing Berth	River Vessel Berth	Number of Seagoing Ships	Number of River Vessels
Small-Scale a	3@3@4@6	3	3	4	6
Small-Scale b	3@4@4@15	3	4	4	15
Medium-Scale a	3@5@5@25	3	5	5	25
Medium-Scale b	4@7@7@25	4	7	7	25
Medium-Scale c	5@10@10@25	5	10	10	25
Medium-to-Large-Scale a	3@7@7@35	3	7	7	35
Medium-Large-Scale b	4@10@10@35	4	10	10	35
Large-Scale	5@15@15@45	5	15	15	45

5.2. Algorithm Effectiveness Evaluation

5.2.1. Model and Algorithm Parameter Settings

The key parameter settings of the model and hierarchical iterative collaborative optimization algorithm in this study are shown in Table 7.

Table 7. HICOA parameter settings.

Parameter Name	Symbol	Value
Maximum Number of Iterations in the Outer Layer	Tmax	200
Maximum Number of Iterations in the Inner Layer	MAXinteractions	1500
Initial Temperature	T0	1200
Cooldown Rate	α	0.95
Maximum Random Seagoing Vessel Removal Ratio	n1	0.5
Maximum Random River Vessel Removal Ratio	n2	0.5
Maximum High-Cost Vessel Removal Ratio	n3	0.4
Disturbance Removal Ratio	ϱ	0.7
Stagnation Threshold	η	100
Better Than Global Optimal Reward	w1	20.0
Better Than Current Solution Reward	w2	10.0
Accept Inferior Solution Reward	w3	1.0
Rejection Penalty	w4	−1.0
Initial Operator Weight	s0	1.0
Seagoing Vessel Port Time Weight	α	1
River Vessel Full Process Time Weight	β	1
River Vessel Lock Passage Time Weight	γ	1

5.2.2. Comparison of Hierarchical Iterative Cooperative Optimization Algorithm and Exact Algorithm Performance

The HICOA was compared with the exact algorithm to verify its solution efficiency and quality. This task was carried out using the commercial solver CPLEX. On each test instance, the HICOA was run independently 10 times, and the mean of the obtained objective function values was recorded. Meanwhile, CPLEX was solved within a set 5 h time limit, and its best feasible solution and GAP1 (%) relative to its own linear relaxation lower bound were recorded.
$$\text{GAP1} = \frac{\text{HICOA optimization objective value} - \text{CPLEX solution objective value}}{\text{CPLEX solution objective value}} \times 100\%.$$
 A negative GAP1 value indicates that the HICOA found a better feasible solution than CPLEX. The comparison re-

sults are shown in Table 8 (“—” indicates that CPLEX did not find a feasible solution within 5 h).

Table 8. Comparison of the HICOA and CPLEX solution results.

Instance	Seagoing Vessel Port Arrival Time Window	HICOA Total Cost (h)	CPLEX Results Total Cost (h)	GAP1 (%)
3@3@4@6	8–10	17.59	17.59 *	0.00
	8–11	17.98	17.98 *	0.00
	8–12	19.73	19.73 *	0.00
	8–13	19.76	19.76 *	0.00
	8–14	18.68	18.68 *	0.00
3@4@4@15	8–10	44.19	44.47	−0.63
	8–11	33.71	36.90	−8.65
	8–12	34.35	34.65	−0.87
	8–13	37.71	37.85	−0.37
	8–14	42.32	42.33	−0.02
3@5@5@25	8–10	83.19	302.35	−72.49
	8–11	74.19	—	—
	8–12	66.50	82.30	−19.20
	8–13	62.67	143.57	−56.35
	8–14	61.10	226.74	−73.05
4@7@7@25	8–10	61.86	—	—
	8–11	60.06	—	—
	8–12	54.49	176.89	−69.20
	8–13	65.15	330.16	−80.27
	8–14	58.30	122.59	−52.44
5@10@10@25	8–10	68.52	281.86	−75.69
	8–11	64.43	209.15	−69.19
	8–12	71.29	—	—
	8–13	66.68	—	—
	8–14	67.66	364.64	−81.45
3@7@7@35	8–10	118.46	—	—
	8–11	102.72	—	—
	8–12	108.51	—	—
	8–13	91.88	—	—
	8–14	92.74	—	—
4@10@10@35	8–10	139.85	—	—
	8–11	126.45	—	—
	8–12	113.31	—	—
	8–13	92.77	—	—
	8–14	114.25	—	—
5@15@15@45	8–10	219.15	—	—
	8–11	220.71	—	—
	8–12	193.46	—	—
	8–13	193.60	—	—
	8–14	187.24	—	—

Note: “—” indicates that CPLEX did not find a feasible solution within 5 h, and “*” denotes that CPLEX found the optimal solution.

In the comparison results in Table 8, in small-scale instances 3@3@4@6, the mean of the objective function obtained by the HICOA is highly consistent with the optimal solution obtained by CPLEX within the 5 h time limit, verifying the correctness of the mixed-

integer linear programming model constructed in this study. Furthermore, the HICOA could achieve an optimization performance comparable to that of the exact algorithm in small-scale problems. The difficulty of solving CPLEX significantly increases with an increase in the problem scale. In medium-to-large-scale and larger instances, CPLEX cannot obtain a feasible solution within the set time limit in most time windows. Even if a solution is obtained, its quality is far lower than that of the HICOA. In contrast, the HICOA could stably obtain feasible solutions on all 40 test cases, and the variance of the results of multiple runs was small, showing good robustness and solution efficiency. The above-mentioned results demonstrate that the HICOA can provide high-quality solutions to large-scale port–lock coordinated scheduling problems within an acceptable computation time. The overall performance of this algorithm was significantly better than that of the commercial solver CPLEX, making it suitable for optimization decisions in complex real-world scenarios.

5.2.3. Performance Comparison of Joint and Independent Scheduling

This section compares the proposed joint scheduling method with traditional independent scheduling methods to demonstrate the practical value of collaborative scheduling between ship locks and ports. The traditional independent scheduling method is a sequential scheduling method that uses a three-stage decomposition strategy for sequential decision making. In the first stage, the downstream lock scheduling problem of river ships is independently solved to determine the departure time of each river ship. In the second stage, this departure time is used as an unchangeable input to independently solve the berth allocation and loading/unloading operation scheduling problem for all ships at the port. Accordingly, the completion time of each ship's operation at the port is determined. Finally, in the third stage, the completion time of the above tasks is used as a fixed input to independently solve the upstream lock scheduling problem of river ships. The total cost of independent scheduling is the sum of the target values of the above-mentioned three stages of the model. The improvement percentage GAP2 (%) of the joint scheduling scheme relative to the independent scheduling scheme in total port time is used as the evaluation indicator:
$$GAP_2 = \frac{\text{Joint scheduling target value} - \text{Independent scheduling target value}}{\text{Independent scheduling target value}} \times 100\%$$
. A negative value indicates that joint scheduling is superior to independent scheduling. The comparative experiment was conducted in two steps: first, the exact optimal solution was obtained on a small-scale instance for theoretical comparison; second, the HICOA was used for approximate solution comparison on large-scale instances.

(1) Performance comparison analysis based on the exact optimal solution

A representative small-scale test case “3@3@4@6” was selected to rigorously verify the superiority of the joint scheduling model at the theoretical level. The commercial solver CPLEX12.8.0 was used to solve the joint and independent scheduling models, with a time limit of 5 h. The comparison results are shown in Table 9.

Table 9. Comparison of joint and independent scheduling results.

Instance	Seagoing Vessel Port Arrival Time Window	Total Cost of Joint Scheduling (h)	Total Cost of Independent Scheduling (h)	GAP2 (%)
3@3@4@6	8–10	17.59	19.55	−11.14
	8–11	17.98	19.63	−9.18
	8–12	19.73	20.51	−3.95
	8–13	19.76	28.55	−44.48
	8–14	18.68	22.33	−19.54

Across all time windows, the exact optimal solution of joint scheduling was better than that of independent scheduling, with the improvement rate reaching up to 44.48% (Table 9). The data show the superiority of joint scheduling.

(2) Comparative analysis of large-scale cases based on the HICOA

This study further used the HICOA to approximate and compare the solutions of larger-scale instances and further verify the advantages of joint scheduling in large-scale practical problems. Each stage of independent scheduling independently calls the HICOA with completely identical parameter settings to ensure a fair comparison. Table 10 records the HICOA solution results and corresponding GAP2 values of joint scheduling and independent scheduling under different scale instances.

Table 10. Comparison of the joint and independent scheduling results.

Instance	Seagoing Vessel Port Arrival Time Window	Mean Objective Value of Independent Scheduling (h)	Mean Objective Value of Joint Scheduling (h)	GAP ₂ (%)
3@3@4@6	8–10	18.52	17.59	−5.02
	8–11	23.67	17.93	−24.23
	8–12	22.66	19.73	−12.93
	8–13	21.72	19.76	−9.03
	8–14	24.08	18.68	−22.44
3@4@4@15	8–10	74.12	44.19	−40.38
	8–11	59.66	33.71	−43.50
	8–12	57.85	34.35	−40.63
	8–13	60.34	37.71	−37.51
	8–14	74.49	42.32	−43.19
3@5@5@25	8–10	192.25	83.19	−56.73
	8–11	173.00	74.19	−57.11
	8–12	154.53	66.50	−56.97
	8–13	142.44	62.67	−56.00
	8–14	125.87	61.10	−51.46
4@7@7@25	8–10	153.95	61.86	−59.32
	8–11	143.46	60.06	−66.22
	8–12	144.43	54.49	−57.75
	8–13	142.34	65.15	−65.67
	8–14	145.55	58.30	−63.33
5@10@10@25	8–10	177.73	68.52	−59.82
	8–11	153.91	64.43	−58.14
	8–12	174.49	71.29	−62.27
	8–13	147.18	66.68	−54.23
	8–14	149.41	67.66	−59.94
3@7@7@35	8–10	291.15	118.46	−57.37
	8–11	304.04	102.72	−57.96
	8–12	256.87	108.51	−56.99
	8–13	267.65	91.88	−65.64
	8–14	252.86	92.74	−59.57
4@10@10@35	8–10	328.05	139.85	−61.45
	8–11	300.75	126.45	−58.14
	8–12	263.44	113.31	−59.15
	8–13	270.00	92.77	−54.70
	8–14	282.56	114.25	−54.71
5@15@15@45	8–10	470.81	219.15	−53.45
	8–11	483.43	220.71	−54.34

8–12	482.96	193.46	−59.94
8–13	448.42	193.60	−56.83
8–14	450.97	187.24	−58.48

The joint scheduling method significantly outperformed the independent scheduling method in all test cases (Table 10). The advantages of joint scheduling become increasingly apparent with increases in problem size and complexity. In small-scale cases, joint scheduling reduced the total port time by 5.02% to 43.50% compared with independent scheduling. Meanwhile, in medium-sized and larger cases, the Gap₂ value typically exceeded 50%, and joint scheduling could minimize the total port time by more than 60%. This result fully verifies the necessity of port–lock coordinated scheduling. The information-segmented decision-making mode of independent scheduling results in a loss of flexibility in adjusting ship arrival times at the port, easily causing resource idleness or congestion. The joint scheduling method proposed in this study achieves the global optimization of resource allocation in the two key links of locks and ports through unified decision making, thereby significantly improving the operational efficiency of the entire intermodal transport system. Additionally, the solution of the joint optimization also demonstrates good stability.

5.3. Sensitivity Analysis

This section selects the medium-scale instance “3@5@5@25@8–12” as the benchmark scenario to extensively explore the influence of model parameter changes on the performance of the joint scheduling system. A sensitivity analysis was conducted from three dimensions: the tightness of the river vessel arrival time, the ratio of the river vessel number to berth number, and the ratio of the seaboat number to berth number.

5.3.1. Influence of the Concentration of River Vessel Arrival Time on System Performance

The concentration of river vessel arrival time directly affects the coordination efficiency between the lock scheduling strategy and the port operations. This experiment gradually widened the river vessel arrival time window from the baseline of 1 h to 5 h, keeping other parameters constant, to explore the influence of the time window width on the total port time. The experimental results are shown in Table 11.

Table 11. Influence of river vessel arrival time window width on total system cost.

Experimental Groups	Width of Time Window for River Vessel Arrival at Lock (h)	Total System Cost (h)
Baseline Scenario	1	66.50
Experimental Group 1	2	59.80
Experimental Group 2	3	71.28
Experimental Group 3	4	82.47
Experimental Group 4	5	82.17

When the time window was widened from 1 h to 2 h, the total system cost decreased from 66.50 h to 59.80 h, a decrease of 10.1% (Table 11). This result indicates that a slightly wider window provides great flexibility for lock scheduling, facilitating the formation of an efficient lock combination. However, when the window was widened to more than 3 h, the total cost rapidly increased. This phenomenon arises because an excessively wide window resulted in a highly dispersed distribution of ship arrival times, increased lock waiting times, and disruption of port operation continuity. Under the conditions of this experiment, a window of about 2 h can effectively balance lock utilization and port operation continuity.

5.3.2. Effect of the Ratio of River Vessel Number to Berth Number on System Performance

The resource matching degree between river vessels and berths is a key factor affecting port operation efficiency. This experiment, with a fixed number of seagoing vessel berths and seagoing vessels, varied the number of river vessel berths to explore the influence of the ratio of river vessels to berths on system performance. The experimental results are shown in Table 12.

Table 12. Influence of the ratio of river vessels to berths on the total system cost.

Instance Configuration	Ratio of River Vessels to Berths	Total System Cost (h)
3@3@5@25	8.33	84.71
3@4@5@25	6.25	70.95
3@5@5@25	5	66.78
3@6@5@25	4.17	64.51
3@7@5@25	3.57	63.13

The experimental results in Table 12 show that the total system cost showed a continuous downward trend with a decrease in the ratio of river vessels to berths. However, the marginal benefit gradually weakened. When the ratio decreased from 8.33 to 5.00, the total system cost decreased from 84.71 h to 66.78 h, a decrease of 21.2%. When the ratio further decreased from 5.00 to 3.57, the cost only decreased from 66.78 h to 63.13 h, a decrease of only 5.5%. The above-mentioned data indicate that under this experimental scenario, maintaining the ratio of river vessels to berths at around 5 can achieve a good balance between resource utilization and system cost, and the cost optimization space brought about by further increasing berths is limited.

5.3.3. Effect of the Ratio of Seagoing Vessels to Berths on System Performance

Seagoing vessels are the core hub of river–sea intermodal transport, and their berth configuration directly affects the operational efficiency of the entire system. This experiment, with a fixed number of river vessel berths and river vessels, varied the number of seagoing vessel berths to explore the influence of the ratio of seagoing vessels to berths on system performance. The experimental results are shown in Table 13.

Table 13. Influence of the ratio of seagoing vessels to berths on the total system cost.

Instance Configuration	Seagoing Vessel Number/Berth Number Ratio	Total System Cost (h)
1@5@5@25	5	79.74
2@5@5@25	2.5	66.20
3@5@5@25	1.67	66.64
4@5@5@25	1.25	66.46
5@5@5@25	1	65.77

The total system cost showed a downward trend with a decrease in the ratio of seagoing vessels to berths, with the rate of decrease leveling off after the ratio dropped to 2.5 (Table 13). When the ratio decreased from 5.00 to 2.50, the cost decreased by 16.8%, the most significant reduction. This notion indicates that when the ratio of seagoing vessels to berths is greater than 2.5, berth resources are severely insufficient. Under the conditions of this experiment, controlling the ratio between 1.5 and 2.5 can effectively achieve efficient resource utilization.

6. Conclusions

This study addresses the problem of poor port–lock connectivity under the direct transshipment model between seagoing and river transport. Moreover, this study explores a port–lock joint scheduling optimization method to alleviate these phenomena, such as vessel congestion, resource idleness, and low system efficiency caused by independent decision making between berths and locks. A mixed-integer linear programming model integrating berth allocation, lock batch scheduling, and lock chamber layout constraints was constructed based on the spatiotemporal coupling characteristics of the port–lock system. A hierarchical iterative collaborative optimization algorithm was designed. The comparative experimental results show that the proposed model could effectively characterize the port–lock collaborative operation process, and HICOA could obtain high-quality feasible solutions more efficiently than CPLEX under the imposed computational time limit, especially for medium- and large-scale instances. Joint scheduling can significantly reduce the total port time and improve the overall operating efficiency of the river–sea intermodal transport system compared with the traditional independent scheduling mode.

Furthermore, the sensitivity analysis based on multi-scale simulations shows that the arrival time window of river vessels and the vessel–berth configuration relationship have a significant influence on system performance. An excessively wide or narrow arrival time window is detrimental to system efficiency and should be kept within a reasonable range. An excessively high ratio of river vessels to berths can create operational bottlenecks, while an excessively low ratio can lead to resource idleness. The ratio of seagoing vessels to berths exhibits a certain threshold characteristic and shows a marginal effect. These conclusions can provide a reference for setting parameters and allocating resources for port–lock coordinated operation.

This study still has certain limitations. Specifically, this study mainly focuses on the port–lock joint scheduling problem of a single port in a deterministic static scenario and has yet to consider the multi-port network environment and random disturbance factors. Future research can be further extended to multi-port river–sea intermodal transport networks and introduce uncertain factors, such as ship arrival time and loading and unloading demand, to construct a dynamic robust scheduling model. Finally, the existing HICOA can be improved by combining intelligent optimization methods, such as reinforcement learning, to improve its solution efficiency and applicability in large-scale complex scenarios.

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

Funding: The authors wish to express their deepest gratitude for the financial support provided by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM804), Guangxi Science and Technology Major Program (No. AA23062021-4), Collaborative Scheduling Optimization of River Ports and Sea Ports (2025320122000418).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Appendix A

Algorithm A1 Lock Scheduling—2D Packing and Batch Allocation

Input:

River vessel set I_j with arrival times $\{T_s^{arr}\}$ ($s \in I_j$); vessel dimensions (SWs: width, SLs: length); lock chamber bounds (CW_{cmax} : max width, CL_{cmax} : max length); safety clearances (δ_w : lateral, δ_l : longitudinal); batch creation cost C_{batch}

Output:

Batch assignment $\{f_{j,k}\}$ ($f_{j,k} = 1$ if river vessel j is assigned to batch k); batch opening times $\{C_k\}$; vessel layout positions $\{x_{pos_{j,k}}, y_{pos_{j,k}}\}$

Phase 1: Generate FCFS Initial Solution

1. Sort all river vessels in non-decreasing order of arrival time T_s^{arr} .
 2. Create the first batch b_1 ; initialize $SW_b = 0$, $SL_b = 0$, $SW_{bmax} = 0$, $SL_{bmax} = 0$. // SW_b , SL_b : cumul. dims; SW_{bmax} , SL_{bmax} : max dims in batch b
 3. For each vessel s in sorted order do
 4. Try to fit s into the current batch b using 2D-Packing-Check(b, s). //see Section 4.1.1
 5. If fit is feasible then
 6. Add s to batch b ; update SW_b , SL_b , SW_{bmax} , SL_{bmax} .
 7. Else
 8. Open a new batch b_{new} ; add s as its first vessel; set $b \leftarrow b_{new}$. //start a fresh lock session
 9. End if
 10. End for
 11. For each batch b , set $C_b \leftarrow \max\{T_s^{arr} \mid s \in b\} + t_c$. // t_c : lock chamber processing time
 12. Initialize batch set $B \leftarrow \{\text{all batches created above}\}$.
-

Phase 2: Cost-Based Batch Merging (Improvement)

13. For each vessel s in sorted order do
 14. $\Delta C_{best} \leftarrow +\infty$; $b_{best} \leftarrow \text{null}$ //track the best merge candidate
 15. For each batch $b \in B$ do
 16. If 2D-Packing-Check(b, s) is feasible then //see Section 4.1.1
 17. $t_{open}(b, s) \leftarrow \max(\max\{T_j^{arr} \mid j \in b\}, T_s^{arr})$ //new opening time if s joins b
 18. $\Delta C_{add} \leftarrow \sum_{j \in b} (t_{open}(b, s) - T_j^{arr}) + (t_{open}(b, s) - T_s^{arr})$ //see Section 4.1.2
 19. If $\Delta C_{add} < \Delta C_{best}$ then $\Delta C_{best} \leftarrow \Delta C_{add}$; $b_{best} \leftarrow b$ end if
 20. End if
 21. End for
 22. $C_{new} \leftarrow C_{batch} + 0$ //cost of keeping s in its own FCFS batch (zero waiting)
 23. If $b_{best} \neq \text{null}$ and $\Delta C_{best} < C_{new}$ then //see Section 4.1.2
 24. Move s from its FCFS batch into b_{best} ; update $C_{b_{best}}$.
 25. End if
 26. End for
 27. Return $\{f_{j,k}\}$, $\{C_k\}$, $\{x_{pos_{j,k}}, y_{pos_{j,k}}\}$.
-

Notes:

2D-Packing-Check(b, s) (Steps 4, 16): Tests lateral placement first ($SW_b = SW_b + SW_s + \delta_w$, $SL_b = \max(SL_b, SL_s)$), and then longitudinal ($SW_b = \max(SW_b, SW_s)$, $SL_b = SL_b + SL_s + \delta_l$); returns feasible if either fits within (CW_{cmax} , CL_{cmax}). Details in Section 4.1.1.

C_{batch} is an algorithm parameter representing the fixed cost of initiating a new lock session; it acts as a surrogate for the lock operation cost coefficients c_u/c_d in the objective function. $C_{new} = C_{batch}$ because vessel s incurs zero waiting in a new batch opened exactly at T_s^{arr} . Details in Section 4.1.2.

Phase 1 ensures that every vessel is assigned a batch (FCFS guarantee). Phase 2 only accepts merges that strictly reduce the total cost, so the final solution is always no worse than the FCFS initial solution.

References

1. Zhang, Y.; Zheng, Q.-Q.; He, L.-J.; Tian, H.-W. Ship traffic optimization method for solving the approach channel and lock co-scheduling problem of the Three Gorges Dam on the Yangzi River. *Ocean Eng.* **2023**, *276*, 114196.
2. Ji, B. Research on Model and Method of Joint Scheduling for Ship Lockage and Berthing in Water Control Project. Thesis, Huazhong University of Science and Technology, Wuhan, China, 2018. (In Chinese)
3. Liu, S.; Zhang, Y.; Guo, W.; Tian, H.; Tang, K. Ship scheduling problem based on channel-lock coordination in flood season. *Expert Syst. Appl.* **2024**, *254*, 124393.
4. Cordeau, J.-F.; Laporte, G.; Legato, P.; Moccia, L. Models and tabu search heuristics for the berth-allocation problem. *Transp. Sci.* **2005**, *39*, 526–538.
5. Agra, A.; Oliveira, M. MIP approaches for the integrated berth allocation and quay crane assignment and scheduling problem. *Eur. J. Oper. Res.* **2018**, *264*, 138–148.
6. Correcher, J.F.; Bossche, T.V.D.; Alvarez-Valdes, R.; Berghe, G.V. The berth allocation problem in terminals with irregular lay-outs. *Eur. J. Oper. Res.* **2019**, *272*, 1096–1108.
7. Corry, P.; Bierwirth, C. The berth allocation problem with channel restrictions. *Transp. Sci.* **2019**, *53*, 708–727.
8. Jia, S.; Li, C.-L.; Xu, Z. A simulation optimization method for deep-sea vessel berth planning and feeder arrival scheduling at a container port. *Transp. Res. Part B Methodol.* **2020**, *142*, 174–196.
9. Korekane, S.; Nishi, T.; Tierney, K.; Liu, Z. Neural network assisted branch and bound algorithm for dynamic berth allocation problems. *Eur. J. Oper. Res.* **2024**, *319*, 531–542.
10. Liu, B.; Wang, X.; Wang, Z.; Zheng, J.; Sheng, D. Modeling and solving the joint berth allocation and vessel sequencing problem with speed optimization in a busy seaport. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *197*, 104089.
11. Wang, T.; Li, T. Ship lock management and dynamic congestion toll for ships. *Ocean Coast. Manag.* **2022**, *230*, 106369.
12. Nauss, R.M. Optimal sequencing in the presence of setup times for tow/barge traffic through a river lock. *Eur. J. Oper. Res.* **2008**, *187*, 1268–1281.
13. Verstichel, J.; Berghe, G.V. A late acceptance algorithm for the lock scheduling problem. In *Logistik Management: Systeme, Methoden, Integration*; Physica-Verlag HD: Heidelberg, Germany, 2009; pp. 457–478.
14. Verstichel, J.; De Causmaecker, P.; Vanden Berghe, G. Scheduling algorithms for the lock scheduling problem. *Procedia-Soc. Behav. Sci.* **2011**, *20*, 806–815.
15. Verstichel, J.; De Causmaecker, P.; Spieksma, F.C.R.; Vanden Berghe, G. Exact and heuristic methods for placing ships in locks. *Eur. J. Oper. Res.* **2014**, *235*, 387–398.
16. Verstichel, J.; De Causmaecker, P.; Spieksma, F.; Vanden Berghe, G. The generalized lock scheduling problem: An exact approach. *Transp. Res. Part E Logist. Transp. Rev.* **2014**, *65*, 16–34.
17. Verstichel, J.; Kinable, J.; De Causmaecker, P.; Vanden Berghe, G. A Combinatorial Benders' decomposition for the lock scheduling problem. *Comput. Oper. Res.* **2015**, *54*, 117–128.
18. Ji, B.; Zhang, D.; Yu, S.S.; Fang, X. An exact approach to the generalized serial-lock scheduling problem from a flexible job-shop scheduling perspective. *Comput. Oper. Res.* **2021**, *127*, 105164.
19. Ji, B.; Zhang, D.; Zhang, Z.; Yu, S.S.; Van Woensel, T. The generalized serial-lock scheduling problem on inland waterway: A novel decomposition-based solution framework and efficient heuristic approach. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *168*, 102935.
20. Deng, Y.; Sheng, D.; Liu, B. Managing ship lock congestion in an inland waterway: A bottleneck model with a service time window. *Transp. Policy* **2021**, *112*, 142–161.
21. Zhao, X.; Liu, S.; Gao, P.; Yu, H. Modelling an improved ship appointment system for lockage operations of waterway transport. *Comput. Ind. Eng.* **2022**, *172*, 108638.
22. Wu, Z.; Ji, B.; Yu, S.S. Modeling and Solution Algorithm for Green Lock Scheduling Problem on Inland Waterways. *Mathematics* **2024**, *12*, 1192.
23. Ji, B.; Yuan, X.; Yuan, Y.; Lei, X.; Fernando, T.; Iu, H.H. Exact and heuristic methods for optimizing lock-quay system in inland waterway. *Eur. J. Oper. Res.* **2019**, *277*, 740–755.

24. Ji, B.; Hou, R.; Yu, S.S. A lock and port resource collaborative scheduling method based on mathematical programming model and heuristic. *Eur. J. Oper. Res.* **2025**, *329*, 476–497.
25. Zhang, X.; Lin, J.; Guo, Z.; Liu, T. Vessel transportation scheduling optimization based on channel–berth coordination. *Ocean Eng.* **2016**, *112*, 145–152.
26. Hao, L.; Jin, J.G.; Zhao, K. Joint scheduling of barges and tugboats for river–sea intermodal transport. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *173*, 103097.
27. Li, S.; Fan, L.; Jia, S. A hierarchical solution framework for dynamic and conflict-free AGV scheduling in an automated container terminal. *Transp. Res. Part C Emerg. Technol.* **2024**, *165*, 104724.

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.