




Article

Integrated Inbound and Outbound Scheduling for Coal Port: Constraint Programming and Adaptive Local Search

Xuan Lu ¹, Yu Zhang ^{1,2}, Lanbo Zheng ^{1,*}, Caiyun Yang ¹ and Junjie Wang ¹

¹ School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China; luxuan-whut@whut.edu.cn (X.L.); sanli@whut.edu.cn (Y.Z.); yangcaiyun@whut.edu.cn (C.Y.); junjie-wang@whut.edu.cn (J.W.)

² Guangdong Inland Port and Shipping Industry Research Co., Ltd., Shaoguan 512100, China

* Correspondence: lanbozheng@whut.edu.cn

Abstract: The effective production scheduling of dry bulk ports is a challenging task that demands meticulous planning, task allocation based on customer requirements, as well as strategic route and timing scheduling. Dry bulk ports dedicated to handling commodities like coal and iron ore frequently engage in blending operations as a strategic imperative to gain market competitiveness. The process of blending coal and ore entails the timely arrival of the requisite raw materials at predetermined locations. Simultaneously, it necessitates the coordination of the sequencing of goods entering and departing the port to align with the operational demands associated with material stockpiles. This paper describes and analyzes an operational scheduling problem encountered by one of the largest coal blending sea ports in China. Specifically, a rich constraint programming model is presented to define operation sequences integrating daily inbound and outbound services provided by the port, minimizing the overall operation time. In order to enhance the practicality of the method, a CP-based adaptive simulated annealing local search algorithm has been designed and developed for the optimization problem. The empirical validation of the proposed method is conducted using both real production data and generated experimental data adhering to specific rules. The results conclusively demonstrate the efficacy and feasibility of the proposed method. This also substantiates its practicality and effectiveness in real-world applications, facilitating efficient production and energy-saving operations for the coal port.



Citation: Lu, X.; Zhang, Y.; Zheng, L.; Yang, C.; Wang, J. Integrated Inbound and Outbound Scheduling for Coal Port: Constraint Programming and Adaptive Local Search. *J. Mar. Sci. Eng.* **2024**, *12*, 124. <https://doi.org/10.3390/jmse12010124>

Academic Editor: Mihalis Golias

Received: 7 December 2023

Revised: 31 December 2023

Accepted: 3 January 2024

Published: 8 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: coal port; integrated scheduling; inbound and outbound operations; constraint programming; adaptive local search; simulated annealing

1. Introduction

Dry bulk cargo constitutes over 70% of global maritime logistics [1], leading to the rapid development of dry bulk ports in response to the expanding dry bulk shipping market. However, unlike the stable growth in dry bulk port freight volume and port electrification upgrades, the production and operation of dry bulk ports still heavily rely on manual experience. The inefficiency of dry bulk port operations results in a substantial disparity between the port throughput and operational efficiency. The port under study is a prominent coal port in terms of automation, having successfully implemented a comprehensive unmanned operation system spanning from resource receiving to cargo exporting. With an annual throughput consistently surpassing 200 million tons for the past three years, it stands as one of the largest ports in China. The automation transformation demands the more intelligent coordination of machine movements to shape the most efficient flow of commodities through the port.

The production system of the port mainly includes an inbound part where coal resources carried by trains to the port are dumped and stacked onto stockyards, and an outbound part where resources in stockyards are reclaimed and blended as coal products

and loaded onto vessels parking at the berths. The stockyards of the port are important places where the inbound and outbound operations interact. When the vessel arrives at a berth, the coal with different characteristics will be taken out from different stockpiles, mixed into the “product coal” in proportion, and then loaded onto the cabin according to the customer’s needs. The vessel then transports the product coal to the destination. This complex process involves two types of operation tasks, i.e., the trains tasks where raw materials are transferred to stockpiles; and the vessel tasks where raw materials are reclaimed from stockpiles and loaded onto vessels. The objective, on both the train side and the vessel side, is to complete their tasks as early as possible. However, it is observed that only one operation task is allowed on a stockpile at the same time. Furthermore, some resources required for loading tasks must wait until the train is unloaded before they become available. These processes require the alignment of simultaneous moves over hundreds of equipment, while respecting operation rules such as precedence, disjunctive and capacity constraints etc., which pose a substantial challenge to the logistics system management. A partial overview of the port under study is outlined in Figure 1.

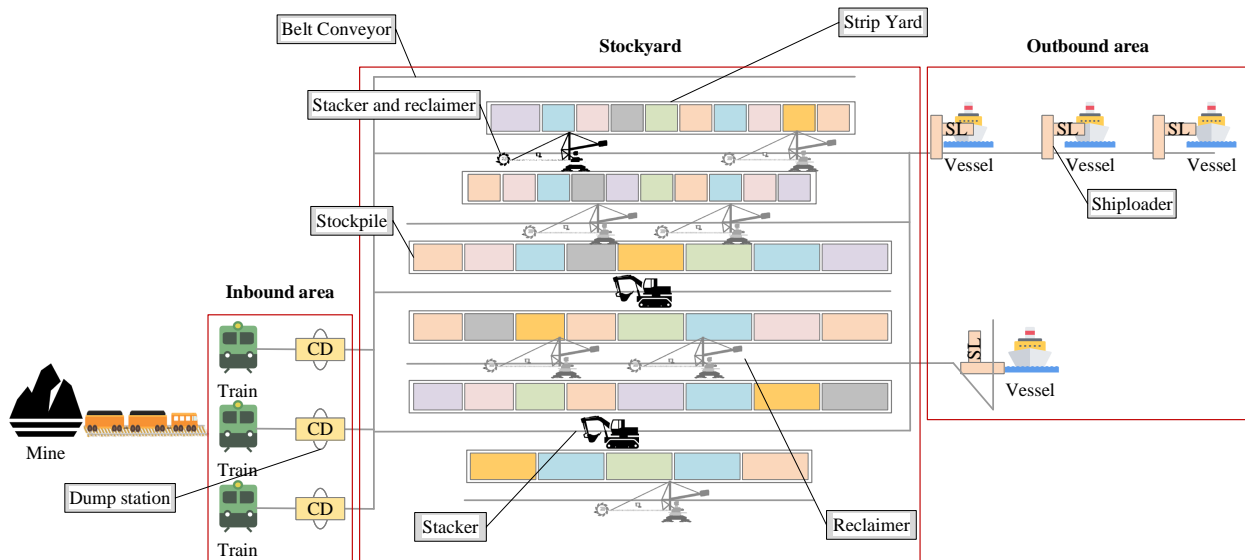


Figure 1. A partial overview of the port under study.

This paper focuses on the integrated scheduling of operation tasks of the coal port. Operation task scheduling is the core of coal port production operations and coal port integrated scheduling. It is very difficult to manage due to the need to make many decisions, such as the coupling between different tasks, the selection of equipment, and the determination of the operation time. Ideally, we hope that each task can be completed as soon as it is generated. However, due to the limited capacities of the resources in the port, this ideal state cannot be achieved.

Our coal port integrated inbound and outbound scheduling approach aims to minimize the overall operation task completion time under the condition of meeting the production constraints of the coal port. Minimizing the operation completion time is used as a proxy for maximizing the throughput and optimizing the production efficiency, which is of great significance for meeting the rising sustainable development and low-carbon production needs of world ports in recent years. Our research is grounded in real-world data, where operation tasks are generated from the actual daily demands for unloading trains and loading ships or datasets created to reflect genuine production conditions. Additionally, equipment information and stockyard conditions are derived directly from the authentic conditions observed at the port.

Considering the complexity and significance of the integrated scheduling of inbound and outbound operations in a coal port, this paper introduces and defines the coal port integrated inbound and outbound scheduling problem based on real coal port production operations. We established a constraint programming model for this problem that incorporates the production operational constraints. Based on this model and the CP solver, we develop an innovative two-stage algorithm that combines constraint programming, adaptive local search, and simulated annealing, leveraging the strengths of these methods. Additionally, this paper conducts a series of multi-scale, multi-parameter experimental analyses for coal port integrated inbound and outbound scheduling problem, providing insights into the problem’s structure and difficulty associated with solving it.

The article is organized as follows. The remainder of this section provides a literature review, followed by a summary of our work in this study. The detailed description and analysis of the integrated inbound and outbound scheduling problem, together with the planning-level issues, are provided in Section 2. Section 3 presents the constraint programming (CP) formulation. Section 4 describes the scheme of the CP-based adaptive simulated annealing local search algorithm. Section 5 contains computational results on randomly generated test instances, as well as on instances derived from real world data. Finally, in Section 6, we point out some directions for further investigation.

1.1. Literature Review

Researchers have shown great interest in the field of dry bulk port planning and operation scheduling, acknowledging its intricate nature and the significance of achieving efficient scheduling in dry bulk ports. In response to these challenges, various optimization techniques have been explored. However, despite the substantial economic benefits and research value associated with addressing the comprehensive scheduling problem of dry bulk ports, the current research in this area remains relatively limited. Presently, more emphasis is placed on investigating unloading scheduling, loading scheduling, and yard scheduling within dry bulk ports. Table 1 provides an overview of the existing literature on the inbound and outbound operations of dry bulk ports, categorized according to the research scope and research methods employed in the articles.

Table 1. Overview of the research on inbound and outbound operation scheduling problems in dry bulk ports.

Citations	Integration	Inbound	Outbound	Land Operations		Stockyard Operations		Shore Operations		Maritime Operations		Method
				TS	SM	SRS	BA	SES	SS	SD		
[2]	-	-	✓	-	✓	✓	✓	-	-	✓	-	MIP + RBH
[3]	✓	✓	✓	-	-	-	-	-	-	-	-	MIP + RBH
[4]	✓	✓	✓	-	-	-	-	-	-	-	-	MIP + RBH
[5]	-	✓	✓	✓	✓	-	✓	-	-	-	-	MIP + RBH
[6]	-	✓	✓	-	-	✓	-	-	-	-	-	MIP + MH
[7]	-	✓	✓	-	-	-	✓	✓	✓	-	-	MIP + RBH
[8]	✓	✓	✓	✓	✓	-	✓	✓	-	✓	-	MH
[9]	-	-	✓	-	-	✓	✓	✓	-	-	-	MIP + BD
[10]	✓	✓	✓	-	-	✓	✓	✓	-	-	-	MH
[11]	✓	✓	✓	-	-	-	-	✓	-	-	✓	MH
[12]	✓	✓	✓	-	✓	-	-	-	-	-	-	MH
[13]	-	✓	✓	-	-	-	-	-	-	-	-	MIP + CG
[14]	-	-	✓	-	-	-	✓	✓	✓	-	✓	MIP + BD
[15]	-	-	✓	-	-	✓	✓	✓	✓	✓	-	MIP + MH
This paper	✓	✓	✓	-	✓	✓	-	✓	✓	-	-	CP + MH

Note: “TS”: train scheduling; “SM”: stockyard management; “SRS”: stacker/reclaimer scheduling; “BA”: berth allocation; “SES”: shoreside equipment scheduling; “SS”: ship scheduling; “SD”: ship deballasting; “MIP”: mixed-integer programming; “RBH”: rule-based heuristic; “MH”: meta-heuristic; “BD”: Benders decomposition; “CG”: column generation; “CP”: constraint programming; ✓: contains this content.

The inbound and outbound operations in bulk ports encompass various production processes that differ based on the nature of the bulk cargo and the mode of transportation involved. The predominant modes of transportation are ship transportation and train transportation, which each require specific inbound or outbound operations. Consequently,

complex scheduling problems arise, including train scheduling, berth allocation, ship scheduling, and inbound and outbound operation scheduling.

The quayside ship scheduling and berth allocation problem has consistently been a prominent issue in the operation of bulk cargo ports. Tang et al. [16] conducted a study focusing on the factors influencing the scheduling of bulk cargo ports. They developed a non-deterministic polynomial (NP) model for the berth scheduling problem in bulk cargo ports and employed a multi-stage particle swarm optimization algorithm to solve the model. Tang et al. [17] established a mixed-integer programming model for the problem of ship unloading and yard allocation in bulk cargo terminals of large iron and steel companies. They effectively solved the model using Benders decomposition, enhanced by various techniques such as effective inequalities, combinatorial Benders cuts, variable reduction tests, and an iterative heuristic procedure. Pratap et al. [7] and Hu [18] both utilized a meta-heuristic algorithm based on the genetic algorithm to optimize the joint allocation of berth and ship unloader configuration in bulk ports. Krimi et al. [19] proposed an efficient heuristic algorithm based on the rolling horizon approach to solve the integrated problem of berth allocation and crane allocation in bulk ports. Peng et al. [20], considering the minimization of port carbon emissions, utilized the multi-objective particle swarm optimization algorithm to achieve the collaborative allocation of bulk port berths and shore power. Gao et al. [13] described the unloading scheduling problem of a bulk cargo terminal in a large steel plant as a flexible shop scheduling problem, established a corresponding mixed-integer programming model, and proposed a column generation method to solve the optimization model. Yang et al. [21] formulated a mixed-integer programming model for scheduling irregular ships in coal ports and utilized the branch pricing algorithm to optimize the scheduling of such ships. Li et al. [22] formulated a multi-objective optimization model that considers multiple traffic constraints for the ship traffic scheduling problem of a shared navigation channel in the context of coal ports. Cheimanoff et al. [23] introduced a solution approach based on the reduced variable neighborhood for the dynamic continuous berth allocation problem in coal ports. For train scheduling, Xu et al. [24] conducted empirical research to optimize the train set strategy for coal ports, utilizing a simulation model to improve efficiency and effectiveness.

The yard, functioning as a connection point and inventory buffer in the inbound and outbound transportation of bulk ports, plays a crucial role in enhancing the overall operational efficiency. Extensive research has been conducted by scholars on yard management methods and technologies, with a specific focus on yard stacking management and yard equipment scheduling. Belov et al. [25] and Savelsbergh and Smith [26] proposed algorithms to optimize the coal stockpile assembly in stockyard operations. These methods, based on large-neighborhood search and tree search, respectively, aimed to enhance the efficiency of coal port operations. Mao and Zhang [27] developed a bulk cargo terminal management system utilizing the Internet of Things and radio frequency identification technology. In the domain of yard equipment scheduling, Angelelli et al. [28] introduced an abstract model for reclaimer scheduling and examined the model's complexity under different conditions, and gave a pseudo-polynomial algorithm for the scheduling problem of two reclaimers with given stockpile locations and a reclaim order. Kalinowski et al. [29] investigated the scheduling problem of reclaimer equipment in the presence of uncertain stacking and reclaiming sequences. They proved that the single equipment scheduling problem is NP-complete under these conditions and proposed a branch and bound algorithm as well as an approximation algorithm to address the problem.

Although previous research has extensively focused on ship scheduling and loading operation scheduling, there is a lack of discussion on the coupling relationship between inbound and outbound operations. However, some of them [14,15] provide valuable descriptions and explanations regarding the problem background and related constraints that are relevant to this paper. By referring to these sources, readers can acquire a more comprehensive understanding of the specific problem context addressed in this paper. Zhang et al. [15] established a multi-objective mathematical model for the loading operation

and vessel traffic scheduling problem in Huanghua Port, and used a heuristic algorithm combining the variable neighborhood search and non-dominated sorting genetic algorithm II to solve the problem. A mixed-integer programming model for berth allocation and shiploader scheduling problem in Huanghua Port was established by Cao et al. [14], and the exact solution is obtained by using the logical Benders decomposition. Similarly, Guo et al. [30] established an integrated scheduling model that manages the scheduling process of vessel traffic and deballasting operations, and used a rule-based scheduling method to solve the multi-objective model.

Despite the significance of the integrated scheduling problem in bulk cargo ports, there remains a relative scarcity of studies addressing this complex problem and its influencing factors. Babu et al. [5] introduced two greedy construction algorithms based on heuristics for yard management, train scheduling, and ship scheduling problems in a coal export port. They successfully implemented ship scheduling and train scheduling while considering port yard management in stages. Rocha De Paula et al. [8] conducted a comprehensive analysis on maximizing the throughput of the coal port system, considering it from the perspective of the entire supply chain. Burdett et al. [10] conducted a comprehensive series of studies on the operations of coal export ports. They tackled the general resource scheduling problem using meta-heuristic algorithms, which encompassed berths, ship loaders, stackers, reclaimers, and stockpile inventory. In subsequent research, they further explored specific port operational constraints, such as coal blending, double recovery, pressure relief tank water delay [11], and introduced new geometry constraints to create a more realistic and detailed model of stacking process [12]. While these papers conducted detailed research on the integrated scheduling problem of inbound and outbound operations and analyzed the impact of a coal blending constraint and the equipment's anti-collision constraint on the difficulty of solving, their approach did not encompass all the operational constraints addressed in this paper. The comprehensive nature of this paper's model, which considers a wider range of constraints, contributes to a more complete and practical solution for the coal port integrated inbound and outbound scheduling problem.

However, current research endeavors have predominantly focused on collaborative scheduling between yards and berths. Boland et al. [2] proposed a dynamic resource allocation and scheduling problem encompassing berths, coal blending spaces, and equipment. They developed an optimization algorithm utilizing mixed-integer programming and heuristic search techniques and validated their approach through example verification. Additionally, Boland et al. [3,4] presented a dynamic network flow model with edge interrupts for preventive maintenance scheduling problems, including coal transportation railways, trains, and port equipment. They also proposed a series of hybrid meta-heuristic algorithms based on linear programming, examining their performance through the experimental analysis of real-world and test cases. Pratap et al. [6] tackled the challenge of integrated yard management and rake scheduling problem in a coal export terminal, specifically addressing conflicts arising from yard resource occupancy during ship loading and rake unloading. In a subsequent study, Pratap et al. [7] shifted their attention to coal import terminals, proposing a decision support model for optimizing port operations. Unsal and Oguz [9] also presented an exact method for effectively solving the complex integrated problem involving berth allocation, reclaimer scheduling, and stockyard allocation in coal ports. Furthermore, Belov et al. [31] tackled the integration of train scheduling, yard management, and ship scheduling within the logistics system planning framework. They formulated a mathematical model independent of a solver, analyzing and comparing its performance in solving mixed-integer programming and constraint programming models.

These studies have made significant contributions to the study of inbound and outbound operations, yard management, ship berths, and traffic scheduling, as well as integrated scheduling in bulk cargo ports. Although the integrated inbound and outbound scheduling problem of bulk cargo ports remains relatively unexplored, these investigations provide valuable insights and methodologies for addressing various aspects of the problem, including resource allocation, preventive maintenance scheduling, and logistics system

planning. Further research in this area can pave the way for comprehensive and efficient solutions to the integrated inbound and outbound scheduling challenges faced by bulk cargo ports.

1.2. Overview of the Work

In this paper, the integrated scheduling technology of coal port inbound and outbound operations based on optimization is developed. The primary goal is to provide a collaborative, efficient and reliable reference scheme for the preparation of the coal port inbound and outbound operation plan. In order to ensure that the integrated scheduling scheme of coal port inbound and outbound operation provided in this paper is meaningful and in line with the actual situation, it is necessary to clearly sort out the coal port inbound and outbound operation process, and capture the key operation characteristics or operational constraints according to the production layout and operational constraints of the coal port, and establish a mathematical model on the appropriate granularity.

This paper aims to fill the gap by specifically addressing the integrated scheduling problem of inbound and outbound operations in bulk cargo ports, considering their interconnected nature and associated challenges. By incorporating insights from existing research, this study provides a unique perspective on the coupling relationship between inbound and outbound operations, enhancing the overall understanding of the field.

This paper focuses on providing a reliable, feasible, and efficient reference scheduling scheme for medium- and short-term scheduling of coal ports. To achieve this goal, the model in this paper utilizes a minute-level granularity for modeling, which introduces additional complexity to the model solution process. Balancing the efficiency of model refinement and the solution process is a major challenge due to the intricate production process and operational constraints involved in coal ports.

The first and most notable contribution of this paper is the comprehensive consideration of coal blending operations and specific operational constraints in the integrated inbound and outbound coal port scheduling problem. Unlike previous studies, this paper accounts for a broad spectrum of practical constraints and the interplay between port inbound and outbound operations, incorporating coal blending operations for ship demand. This approach ensures a more realistic and accurate representation of actual port operations. At the same time, this paper innovatively considers the impact of inbound and outbound scheduling operations on the coal port integrated scheduling problem, and summarizes it as an academic research problem.

The second contribution involves creating a constraint programming model that accurately represents real-world scenarios and devises a solution algorithm that optimizes the balance between computation time and solution efficiency. Acknowledging the distinctive problem structure inherent in the comprehensive scheduling challenge of coal port inbound and outbound operations, we leverage a constraint programming methodology to address it. Introducing an innovative two-stage algorithm that integrates constraint programming (CP), adaptive local search (ALS), and simulated annealing (SA), we enhance the solver's ability to effectively escape local optima. This optimization process improves both the solution results and solution time for the model, proving particularly crucial when tackling large-scale instances where efficiency is paramount.

The third contribution lies in the practical application of the proposed algorithm to real-world scenarios, affirming its viability and efficiency in actual coal port operations. Through extensive experimentation and analysis, this paper unveils the distinctive features of the problem and highlights the advantages of the proposed approach across diverse operational conditions. Furthermore, a comparative analysis is presented, pitting the two proposed solving methods against the variable neighborhood search (VNS) and simulated annealing (SA) algorithms, effectively showcasing the superior efficiency of the algorithm introduced in this paper.

2. Problem Description: The Coal Port Integrated Inbound and Outbound Scheduling Problem

This paper aims to develop the technology based on mathematical optimization for coal port integrated inbound and outbound scheduling. The main goal is to provide a reliable and effective scheduling scheme for minimizing the port operation time. Considering that this technology will be used to formulate the medium-term production scheduling scheme of coal ports (usually a two- to three-day scheduling scheme), it is very important to accurately and precisely describe the characteristics and granularity of the problem. These characteristics determine the problem structure and difficulty of solving the coal port integrated inbound and outbound scheduling problem.

As the junction point of inbound and outbound operations, the coal port yard is an important buffer for balancing the supply and demand differences. As mentioned in the preface, different ports have different storage yard management modes. The port studied in this paper adopts a proprietary storage yard management mode, in which the storage yard space is divided into several parallel strip storage yards, each strip storage yard has several stockpiles with different types and stocks. There are several stackers and reclaimers on both sides of each strip yard, and the equipment moves along the tracks on both sides of the yard. Only one piece of equipment is allowed to stack or reclaim at the same time for each stack. A scheme of the stockyard is given in Figure 2.

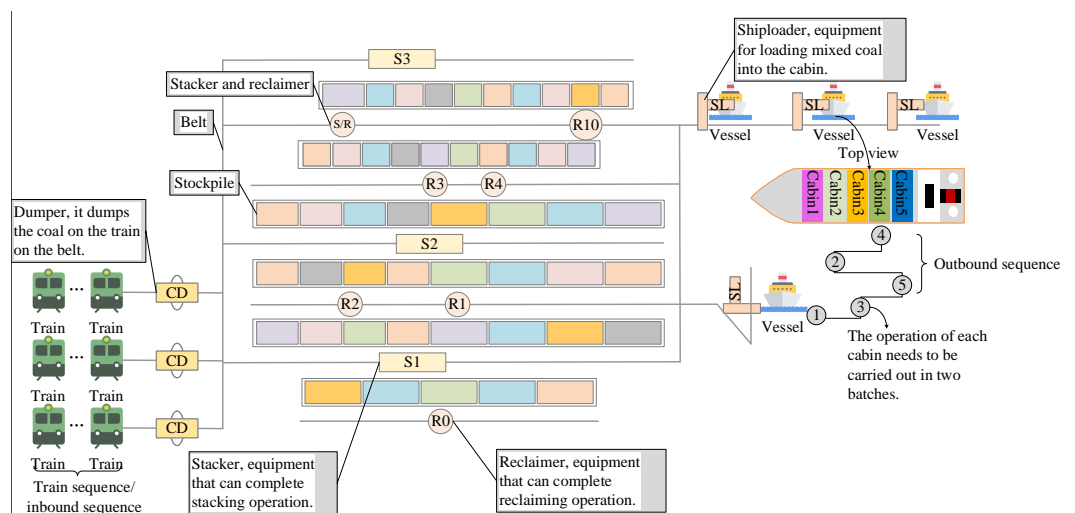


Figure 2. A scheme of the coal port inbound and outbound operation instructions.

As shown in Figure 2, after the raw coal mined in the mine is transported to the port by train, the coal loaded on the train is unloaded to the belt line by the car dumper. Then, the coal is transported to the designated stockpile by belt and loaded onto the stockpile by stacker. After the ship arrives at the port, it will inform the coal manager of the ship's information and the type and quantity of goods required. As coal is a mixed product, coal managers need to formulate different coal blending schemes according to each kind of goods required by customers, so as to better meet customers' requirements for coal price, gray scale and other properties. The coal blending scheme describes which kinds of raw coal are required to be mixed into product coal and the mixing proportion. After berthing, the ship can begin the outbound operation, that is, according to the ship's cargo demand and coal blending scheme, take out the coal from the designated stack by reclaimer, transport it to the shiploader by belt, and then load it to the designated cabin by the shiploader. Because the ship needs to keep balance during outbound operations to prevent a rollover, the ship must be loaded in a certain order and batch. Each batch of each cabin will generate 1–2 operation tasks according to the coal blending scheme of the product coal required by the cabin. The goal of scheduling is to complete the inbound and outbound tasks given by the port manager as soon as possible under certain production constraints.

The production management process of the port under study is divided into two stages: planning and scheduling. In the inbound operation link, the port planner specifies the stockpiles that can be unloaded for the arriving trains during the planning period, as well as the dumper and operation sequences of the inbound operation according to the current production situation, and then generates the inbound operation table. In the outbound operation link, the port planner will formulate several optional coal blending schemes for the product coal required by each cabin of the ship according to the ship's demand and storage conditions, define the operation requirements, and generate the outbound operation table. The operation requirements mainly refer to the operational constraints or operation priorities to be considered during the outbound operation, such as which coal blending scheme is preferred. In the scheduling phase, the port dispatcher will first select the operation stockpile of each operation task according to the operation tables given by the planner, and then select the equipment to perform the operation task and the operation sequence according to the equipment accessibility relationship on the basis of the comprehensive consideration of the storage yard stacking and equipment occupation. In this study, the inbound/outbound operation that determined the operation stockpile is called the operation task, and a complete operation path composed of various operation equipment selected to complete the operation task is called an operation stream. Each operation task can only be completed by one operation stream. A description of operation tasks and operation streams is shown in Figure 3.

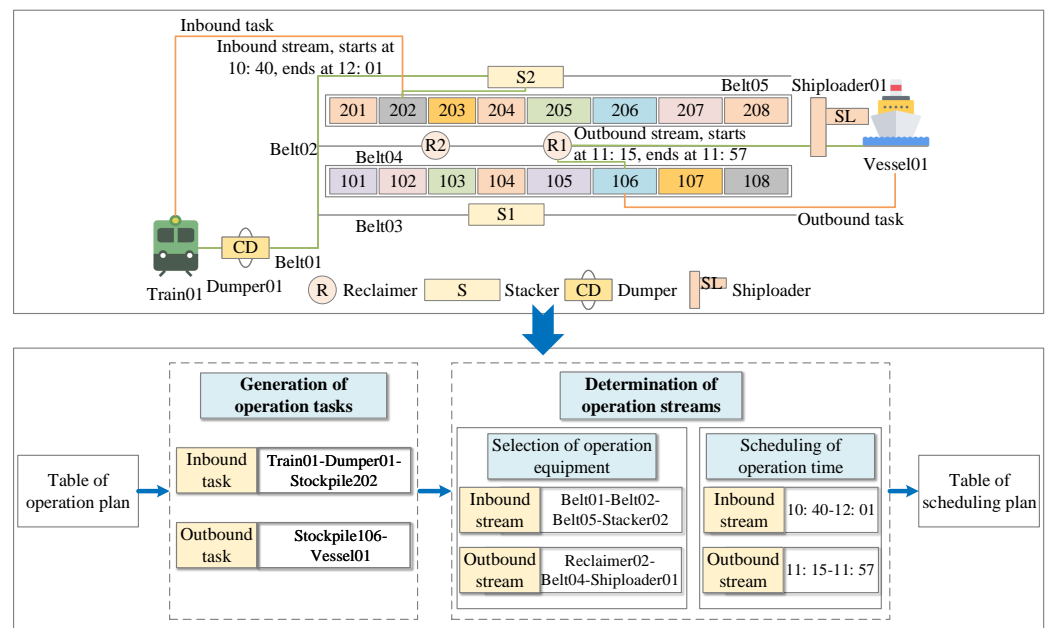


Figure 3. A description of operation tasks and operation streams.

Because the coal port managers are often troubled in the task scheduling stage in the management process, the coal port integrated inbound and outbound scheduling problem (CPIIOS) in this paper assumes that only the generated tasks need to be scheduled. When scheduling tasks, this paper will consider the equipment capacity, equipment moving speed and yard capacity. At the same time, this paper assumes that all operation tasks can start immediately as long as they do not violate the operational constraints.

This paper adopts a minute-level modeling approach to represent the CPIIOS problem, matching the accuracy used by port managers in generating inbound and outbound scheduling schemes and actual operations. This finer granularity provides a more realistic reflection of the challenges faced in coal port scheduling and meets the requirements for medium-term coal port inbound and outbound scheduling. The CPIIOS problem necessitates the consideration of various constraints arising from port layout and production processes:

- Operation stream matching and reachability constraint. An operation task can only be completed by one operation stream, and the operation stream performing the operation task must meet the equipment accessibility.
- Coal blending constraint. If a ship has coal blending requirements during outbound operation, the workload of the operation tasks must meet the proportion requirements of the coal blending scheme given by the port planner.
- Operation streams do not overlap constraint. When an inbound/outbound operation stream starts to perform an operation task, the resources on the operation stream cannot be released before the end of the task. Therefore, operation tasks using the same operation stream cannot overlap in operation time.
- Operation precedence constraint. When determining the start and end time of the operation stream for the inbound/outbound task, operation sequence requirements caused by the train/ship arrival sequence, loading sequence requirements, and other factors must be observed. For example, if the cabin loading sequence of a ship with five cabins is 4–2–5–3–1 and it operates in two rounds. Each round of operations shall be loaded in the sequence of 4–2–5–3–1, and the next round of outbound operations can be carried out only when the operation volume of each cabin reaches the set target of the first round.
- Stockpile capacity constraint. Each stockpile can only accept one piece of equipment for operation at the same time. Therefore, stockpiles can be considered as a special type of equipment added to the operation stream. In addition, the raw coal inventory on the stockpile shall be kept between 0 and the maximum inventory of the stockpile during the process of consumption (outbound operation) and replenishment (inbound operation).
- Equipment anti-collision constraint. Pieces of equipment operating on the same track shall be kept at a certain safe distance. No equipment collision or equipment crossing is allowed.

To sum up, it can be found that the CPIIOS problem in this paper is the process of selecting and planning the corresponding operation streams for the generated operation tasks, reasonably scheduling these operation streams and arranging the operation time under specific constraints.

3. The Optimization Model

Based on the above description of the CPIIOS problem, this section states that the optimization model in the constraint programming inspired modeling language OPL [32]. This paper presents the model in mathematical notation, using nonlinear constructs such as logical operators, variable subscripts, and some special constraints in constraint programming languages, such as cumulative and no-overlap. The model is linearized and solved by the CP Optimizer [33].

An important basic concept in the model is the interval variable. Interval variables have the attributes of present, start, end, and length (usually length = end – start). Interval variables are usually used to model quantities with interval characteristics. Their unique existence attributes make the existence state of variables optional. The constraint programming language also provides a large number of functions related to interval variables. Most of these functions have intuitive and concise meanings. For example: $start(a)$ represents the starting point of interval variable a ; $end(a)$ represents the end point of interval variable a ; and $presence(a)$ represents whether interval variable a appears in the current solution.

Unless otherwise specified, all times are in minutes, the lengths are in meters, and the weights are in tons. Moreover, all values are assumed to be integers.

3.1. Objective Function

The default objective function of this paper is to minimize the sum of the completion time of the latest inbound operation task and the latest outbound operation task, $\min(\max\{end(h)|h \in H\} + \max\{end(v)|v \in V\})$, as the embodiment of maximizing the

port throughput. In the following paper, we also consider maximizing the average equipment utilization and minimizing equipment load imbalance as the optional objective functions.

3.2. Overall System Constraints and Bounds on Variables

Overall constraints mainly include operation stream selection and operation time constraints for operation tasks.

Only one operation stream can be selected for each operation task (that is, each operation stream interval for the same operation task has one and only one state that is present), and the start and end times of the operation stream interval are the same as that of the operation task (*alternative(a, B)* indicates that there is only one variable in set *B* that is equal to *a*):

$$\text{alternative}(h, F(h)), \forall h \in H \tag{1}$$

$$\text{alternative}(v, F(v)), \forall v \in V \tag{2}$$

The start time of the operation stream shall not be earlier than its earliest theoretical start time, nor later than its latest theoretical start time:

$$|T| \geq \text{start}(f) \geq \text{est}_h^{\text{unload}}, \forall f \in F(h), \forall h \in H \tag{3}$$

$$|T| \geq \text{start}(f) \geq \text{est}_v^{\text{load}}, \forall f \in F(v), \forall v \in V \tag{4}$$

Since the actual production process constraints such as the unloading sequence on the dumper, the loading batch and sequence of outbound operation need to be considered, the operation task time needs to meet the operation task precedence constraints, that is, the start time sequence of the operational task needs to be consistent with the operation sequence required by the production process (*endBeforeStart(a, b, c)* indicates that *a* cannot start until *b* ends *c* time units):

$$\text{endBeforeStart}(f, f', \text{lead}), \forall f \in F(h), \forall f' \in F(h'), \left\{ (h, h') \mid \text{ord}_h^{\text{unload}} > \text{ord}_{h'}^{\text{unload}}, \forall h, h' \in H \right\} \tag{5}$$

$$\text{endBeforeStart}(f, f', \text{lead}), \forall f \in F(v), \forall f' \in F(v'), \left\{ (v, v') \mid \text{ord}_v^{\text{load}} > \text{ord}_{v'}^{\text{load}}, \forall v, v' \in V \right\} \tag{6}$$

Operation streams using the same production resource cannot overlap each other in operation time (*sequence(A)* arranges the elements in set *A* in order, *noOverlap(A, lead)* indicates that the elements in set *A* do not overlap with each other and keep a certain interval):

$$\text{noOverlap}(\text{sequence}(F(m)), \text{lead}), \forall m \in M \tag{7}$$

It should be noted that all variables in the constraint programming model are interval variables, so they have an optional existence status (when the variable status is absent, this means that the variable is not included in the final result, otherwise it is included):

$$\text{interval } f, \text{optional}, \forall f \in F(h), \forall h \in H \tag{8}$$

$$\text{interval } f, \text{optional}, \forall f \in F(v), \forall v \in V \tag{9}$$

Following the above constraints, we can obtain a scheduling scheme that meets the production process requirements of general coal ports, as shown in Figure 4. In the figure, each rectangle represents a distinct operation task, with the text label indicating the corresponding operation stacking. The various line trajectories depict the movement path of the operation equipment.

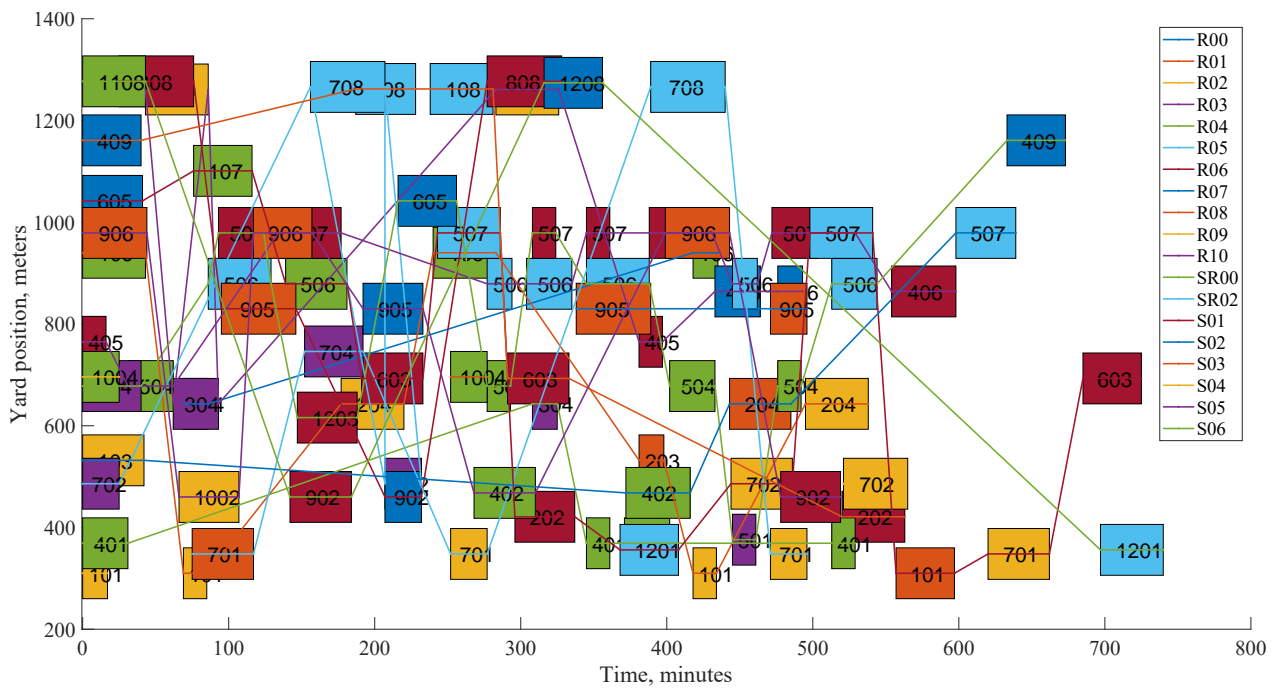


Figure 4. A general scheduling scheme for the coal port.

3.3. Stockpile Operational Constraints

The coal port yard is a place to stack and store the coal unloaded by inbound operations according to the type of coal, so that the coal can be taken out by outbound operations. It is the intersection and buffer of the inbound operation and outbound operation. As depicted in Figure 5, the stockpile operational constraints ensure that the stacking inventory adjusts as the reclaiming operation reduces it and the stacking operation increases it during operations, always maintaining the inventory level between 0 and the maximum allowed.

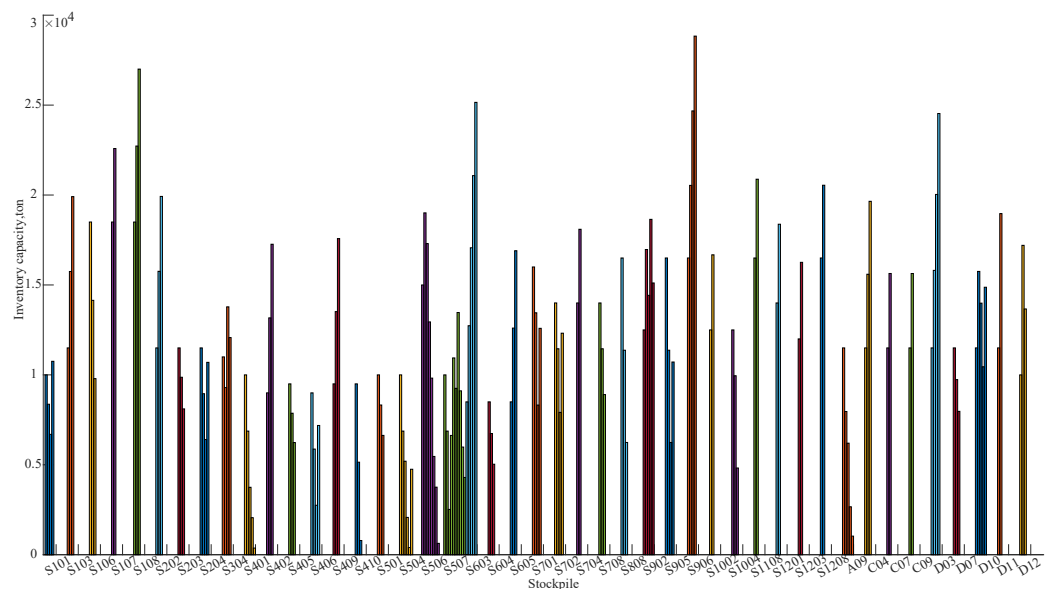


Figure 5. A schematic example of change in stockpile inventory of coal port.

The stockpile inventory in the stockyard will increase or decrease with the inbound and outbound operations. When ignoring the stockpile inventory constraint in the process of operation, it can be found that this problem will be closer to the scheduling problem. The analysis of the solution to this kind of scheduling problem will also be mentioned below. Therefore, it is necessary to ensure that the stockpile inventory is always kept within a

reasonable range during the operation, that is, the stock cannot be 0 or exceed the maximum stock (cumulative constraint uses pulse function to reflect the change of stockpile inventory, and uses the alwaysIn constraint to constrain the change of stockpile inventory):

$$i^s = \sum_{f \in F(s)^{unload}} pulse(f, siz_f^{unload}) - \sum_{f \in F(s)^{load}} pulse(f, siz_f^{load}), \forall s \in S \tag{10}$$

$$alwaysIn(i^s, 0, cap_s), \forall s \in S \tag{11}$$

3.4. Stacker and Reclaimer Operational Constraints

The stacking and reclaiming process of each operation task can be completed by the corresponding equipment on both sides of the strip yard. In order to accurately describe the inbound and outbound process, the moving speed of the equipment can be set according to the actual operation parameters (30 m/min). Considering that there are usually 1–2 types of equipment on the equipment track, there are two types of equipment constraints to consider.

The first is the equipment operation gap constraint: the time required for the equipment to move from the current operation position to the next operation position when completing two consecutive operation tasks.

$$max[end(f_1) - start(f_2), end(f_2) - start(f_1)] \times spd_m \geq |pos_{s_{f_1}} - pos_{s_{f_2}}|, \forall f_1, f_2 \in F(m), \forall m \in R \tag{12}$$

The other is the equipment's anti-collision constraint: equipment on the same track cannot pass each other.

$$max[end(f_1) - start(f_2), end(f_2) - start(f_1)] \times \min(spd_{m_1}, spd_{m_2}) \geq |pos_{s_{f_1}} - pos_{s_{f_2}}| + dis, \tag{13}$$

$$\forall f_1, f_2 \in F(m), \forall m_1, m_2 \in R(g), \forall g \in G$$

Figure 6 shows an example of equipment movement trajectory. In the figure, reclaimers R03 and R04 are situated on the same track on the south side of Strip Yard 4, while stacker-reclaimer SR00 and reclaimer R10 are positioned on the track on the north side of Strip Yard 4. The ordinate in the diagram indicates the equipment's location on the strip yard and represents the name of the stockpile corresponding to each location. For example, S401 indicates stockpile 401 on Strip Yard 4. As can be seen from the figure, equipment belonging to the same track always maintains a certain safety distance and a certain relative position to avoid equipment collision or equipment crossing. And there is only one piece of equipment operating on the same stack at the same time. These findings affirm the importance and effectiveness of the equipment operational constraints presented in this paper.

3.5. Special Operational Constraints

The unique production process of the port under study also brings special production process constraints to this problem. In contrast to the Hunter Valley coal supply chain studied by [31], which first mixes coal according to the ship's demand into a stockpile and then reclaims from it, in the port under study, the raw coal is stored in different stockpiles. After the ship arrives at the berth, it is taken from the stockpile according to the ship's demand, then mixed on the belt line, and finally loaded into the cabin.

In order to meet the requirements of the coal blending process, the outbound operation tasks belonging to the same coal blending scheme must start at the same time:

$$start(f_1) = start(f_2), \forall s(f_1), s(f_2) \in b, \forall b \in B \tag{14}$$

Figure 7 is an example of the outbound operations of a cabin of a ship in the berth. The figure depicts the scheduling of outbound operation tasks, with the x-axis representing the operation time and the y-axis representing operation cabins. Each block represents an outbound operation task, with its width proportional to the task's workload. The text on

each block indicates the specific outbound operation stream, such as “S106–RC01–SL02”, which signifies that the operation task is handled by reclaimer RC01 on stacking S106 and loaded by shiploader SL02. It can be seen that the operation tasks belonging to the same coal blending scheme need to start at the same time to ensure that raw coal can be uniformly blended according to the coal blending ratio.

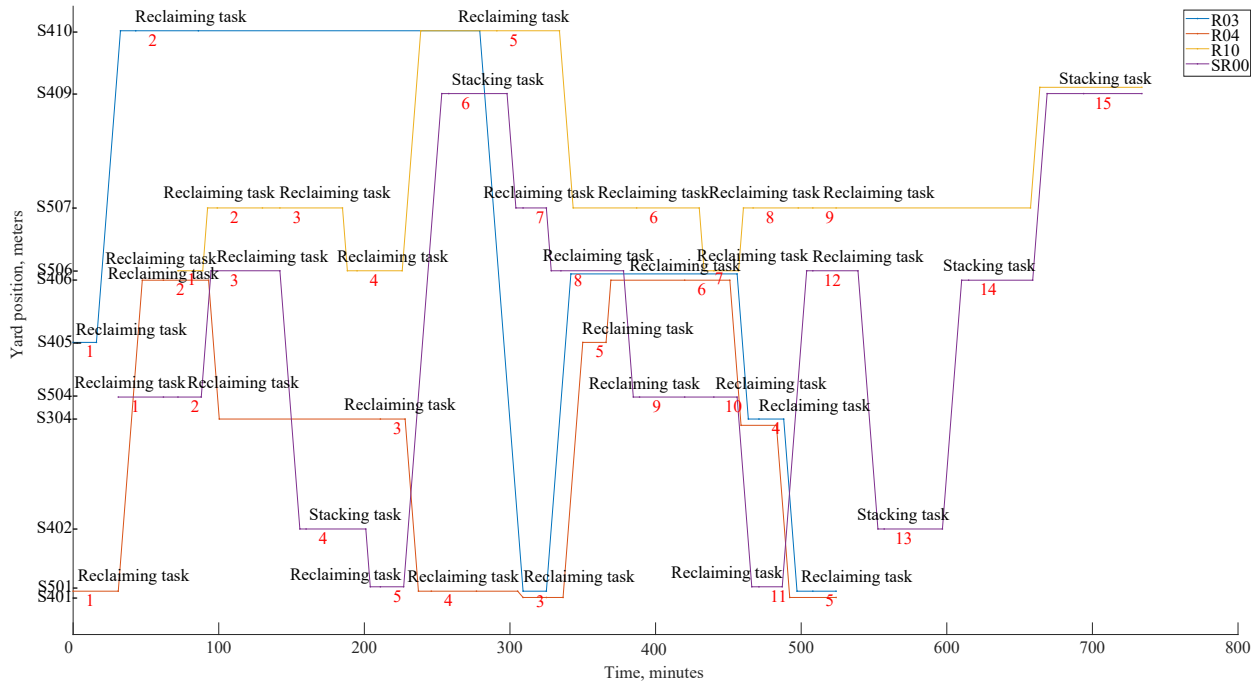


Figure 6. A schematic example of equipment movement track of R03, R04, R10, and SR00.

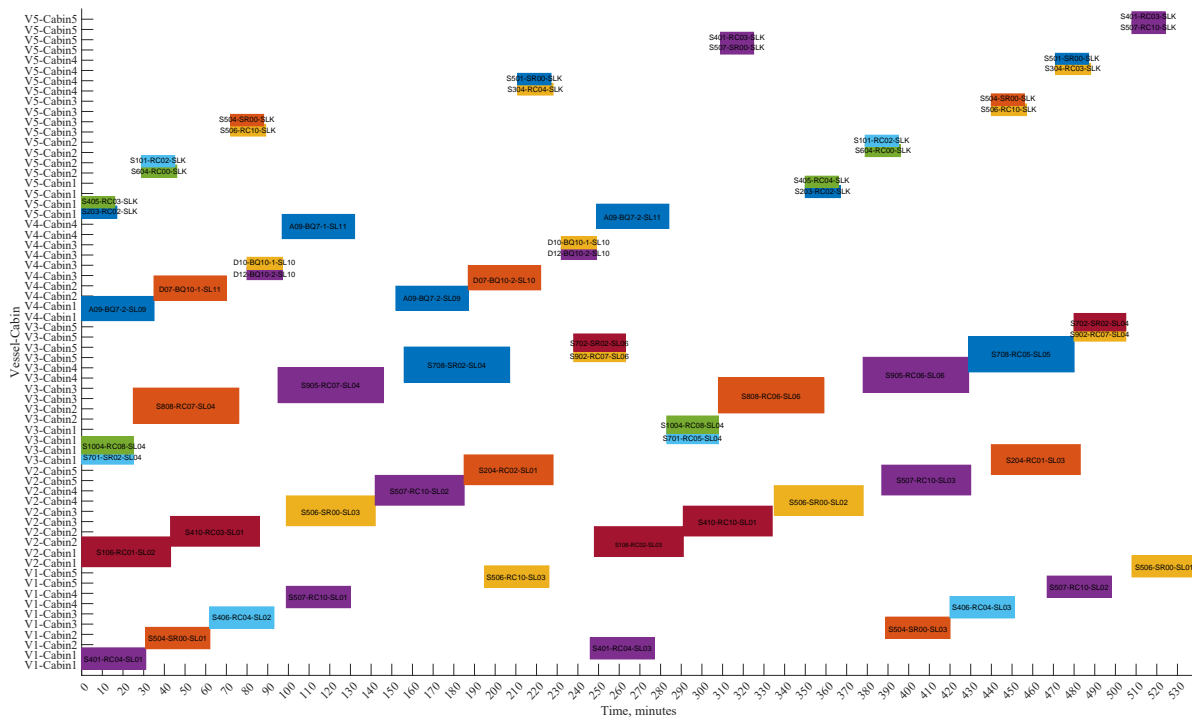


Figure 7. A schematic example of outbound operations of a cabin.

3.6. Redundant Constraints

Since constraint programming plays a pivotal role in optimization by virtue of its capacity for reducing variable domains through constraint propagation, it is of paramount importance to exhaustively enumerate all requisite constraints. Furthermore, the judicious use of constraint recombination, global constraints, redundant constraints, and surrogate constraints stands out as a strategic approach that can substantially amplify the efficacy of the solution process [34].

Given that the issues addressed in this study encompass resource allocation and equipment selection, we devised resource constraints for various equipment types, such as stackers and reclaimers. These constraints serve to restrict the cumulative occupation of each equipment category during specific time intervals, ensuring it does not surpass the total quantity of such equipment available. Experiments show that these redundant constraints can bring significant efficiency gains in accelerating the convergence of the constraint programming model.

$$i^m = \sum_{f \in F(m)} pulse(f, 1), \forall m \in M \tag{15}$$

$$alwaysIn(i^m, 0, cap_m), \forall m \in M \tag{16}$$

4. The Methodology: Heuristic Strategies to Improve the Solution

The presented model can be solved by a CP solver to obtain the optimal or near-optimal solution. Nonetheless, when dealing with larger problem scales or increasingly complex structures, the solution performance of the CP solver exhibits gradual deterioration. Recognizing the inclination of heuristic and meta-heuristic algorithms to become ensnared within local optima when addressing intricate optimization challenges [35,36], this research offers a comprehensive explanation on the configuration of CP solver parameters and strategic solutions to enhance its efficiency. Additionally, it introduces an enhanced two-stage solution algorithm that amalgamates ALS, SA, and the CP solver, with the specific aim of avoiding such suboptimal solutions. It incorporates several heuristic rules to strike a balance between the global search and local search capabilities. These rules encompass the use of roulette wheel rules, adaptive operator selection probabilities, and neighborhood operators tailored to the specific problem structure.

4.1. Outline of Proposed Algorithms

The CP Optimizer, when employed with its default search parameters, operates with a predefined search algorithm and constraint propagation process. In cases where the CP Optimizer detects inconsistency during the predefined constraint propagation, it will engage in backtracking until it finds a feasible solution. In constraint satisfaction problems with multiple or numerous feasible solutions, the primary challenge faced by the CP Optimizer is not in finding a feasible solution, but in optimizing the objective function [37]. The CP Optimizer can swiftly generate feasible solutions for such problems, yet minimal backtracking may hinder its ability to escape local optimal solutions. Consequently, as depicted in Figure 8, this study introduces an approach that utilizes the CP Optimizer to address the CPIIOS problem. Additionally, it incorporates a framework that amalgamates SA and ALS algorithms to augment the CP Optimizer’s capacity for exploring a broader spectrum of viable solutions.

The ALS algorithm is an advanced version of the LNS and VNS algorithms, recognized for its efficiency within neighborhood search algorithms [38]. Compared with the general neighborhood search algorithm, the adaptive neighborhood search algorithm can dynamically manage the neighborhood operators based on their historical performance as the search progresses [39]. Additionally, we incorporate the SA algorithm for accepting or rejecting new solutions. Developing an adaptive simulated annealing local search (ASALS) algorithm for complex constraint satisfaction problems involves significant design and programming efforts. By adopting a solution structure akin to that of constraint programming

Search type: The search types provided by the CP Optimizer include depth-first search, restart search, and multi-point search. Restart search has the best solution efficiency in this model.

Inference level: Changing the inference level for constraint propagation can alter the inference level and calculation time of constraint propagation. With a large number of no-overlap constraints in this paper's model, setting its inference level to the medium can strike a balance between solving efficiency and time.

Workers: The CP Optimizer uses multiple workers for parallel searches. While multi-threading can accelerate the search speed, it also takes up some search time. In this constraint programming model, setting the number of workers to eight can achieve optimal search efficiency.

The CP Optimizer offers user-defined variable search order strategies, including variable search orders and variable value search orders. However, in the model presented in this article, the custom search order strategy based on heuristic rules did not show significant improvements.

Besides the parameter settings and search strategies discussed previously, the CP Optimizer offers further solution optimization by adjusting parameters such as temporary relaxation, random seed, and failure-directed search. Additionally, custom constraint propagators can achieve more consistent constraint propagation with the problem structure. However, due to their limited impact on the efficiency of the constraint programming model presented in this paper, these parameters will not be further examined in this study.

4.3. First Phase of ASALS-CP Algorithm

In the first phase, the ASALS algorithm is introduced, which dynamically adjusts the selection weights of various neighborhood operators based on their historical performance during execution, thus expediting the optimization process. As depicted in Algorithm 1, the initial solution is first derived through the CP solver, with a further detailed solution strategy found in Section 4.2. The initial weight for each neighborhood operator is initialized by setting it to 1. Once the initial solution and operator weights are established, the algorithm proceeds to its iterative phase, continuing until the iteration count (*iter*) is more than *MaxIter*. During each iteration, a neighborhood operator is randomly chosen based on its assigned weight and executed for a duration of τ seconds or all the neighborhood nodes are traversed (*node* > *Node*). To account for variations in CPU time among different neighborhood operators, this study assigns equal running time to each operator, rather than an equal number of iterations, in order to facilitate a more precise calculation of the adaptive operator selection probabilities.

Furthermore, the SA algorithm is used to accept or reject the current solution (S_{temp}) obtained from the neighborhood search. If the objective function value of S_{temp} (C_{temp}) is lower than the objective function value of the best solution found so far (C_{best}), S_{temp} replaces both S_{best} and S . Additionally, if the condition $random(0, 1) < exp\left(\frac{C_S - C_{temp}}{\theta}\right)$ is met, S_{temp} is also accepted as the current solution S .

4.3.1. Representation

The accuracy of solutions in hybrid algorithms significantly relies on the solution transfer between various components within the algorithm, constituting a crucial step in the optimization process of the meta-heuristic algorithm. In this study, the configuration of the solution is represented by the allocation of operation tasks to specific operation equipment and resources, along with the corresponding operation time. The details regarding the transfer of solutions between different algorithms are comprehensively elucidated in Section 4.4.

4.3.2. Initial Solution

The initial feasible solution to the problem is obtained using the CP Optimizer to solve the constraint programming model, with its search parameters and strategies detailed in the previous section. Due to variations in problem structures and scales, the time required for the CP solver to achieve the initial solution may differ. To strike a balance between obtaining a high-quality initial solution and minimizing the computation time, an adaptive adjustment strategy for the solver solution time is designed in this paper. Specifically, the solver's current solution state is monitored every second. The solver stops and outputs the current solution as the initial solution for the ASALS algorithm when the current solution remains a feasible solution for five consecutive times without changes.

Algorithm 1: ASALS

Data: parameters of instances, parameters of ALS, parameters of SA
Result: optimal solution

- 1 Construct an initial solution S_0 using CP solver;
- 2 $S, S_{temp}, S_{best} \leftarrow S_0$
- 3 Initialize weights of neighborhood operators;
- 4 $iter \leftarrow 1$
- 5 **while** $iter \leq MaxIter$ **do**
- 6 select a neighborhood operator n randomly according to the adaptive probabilities;
- 7 $node \leftarrow 1$
- 8 **while** $duration < \tau$ and $node \leq Node$ **do**
- 9 Apply the operator n to find a neighbor solution S' ;
- 10 Check whether S' is feasible;
- 11 **if** S' is feasible **then**
- 12 Calculate objective function value C' of S' ;
- 13 **if** $C' < C_{temp}$ **then**
- 14 $C_{temp} \leftarrow C'$
- 15 $S_{temp} \leftarrow S'$
- 16 **end**
- 17 **end**
- 18 $node \leftarrow node + 1$
- 19 **end**
- 20 $\theta \leftarrow \theta \times \mu$
- 21 **if** $C_{temp} \leq C_{best}$ **then**
- 22 $C_{best} \leftarrow C_{temp}$
- 23 $S, S_{best} \leftarrow S_{temp}$
- 24 **else if** $random(0, 1) < exp\left(\frac{C_S - C_{temp}}{\theta}\right)$ **then**
- 25 $S \leftarrow S_{temp}$
- 26 **end**
- 27 Update the weights and adaptive selection probabilities of neighborhood operators;
- 28 $iter \leftarrow iter + 1$
- 29 **end**

4.3.3. Adaptive Control Mechanism

The primary advantage of the ALS algorithm presented in this paper lies in its adaptive control mechanism, which can dynamically adjust neighborhood operator weights based on specific instances and each step of the algorithm. We employ a roulette wheel selection mechanism to choose neighborhood operators with varying weights. Initially, the weight of each operator is set to $w_n = 1, \forall n \in N$. After each iteration, the operator's weight is

updated based on its success rate r_n in previous iterations. The selection probability p_n of an operator can be calculated according to its weight, as shown in the following formula. This paper follows the operator selection probability and weight update approach detailed in [41]. A minimum selection probability of 2% is set to guarantee that each operator has at least a 2% probability of being selected, preventing operators from being eliminated and ensuring operator diversity while maintaining computational efficiency.

$$p_n = p^{\min} + \frac{w_n}{\sum_{n \in N} w_n} + (1 - |N|p^{\min}), \forall n \in N \tag{17}$$

The weight update is defined as follows. A reaction factor $\alpha = 0.5$ is employed to regulate the impact of weight adjustment on changes in the operator’s success rate in the previous iteration [42]. A higher (or lower) reaction factor results in a greater (or lesser) influence of the latest iteration on weight adjustment. To give preference to operators that are frequently used, we calculate the success rate of operator n in the last iteration and divide it by the number of times that the operator has been applied, denoted by η .

$$w_n = (1 - \alpha)w_n + \alpha \frac{r_n}{\eta}, \forall n \in N \tag{18}$$

The success rate calculation method is as follows. Initially, the success rate of each operator is set to zero at the start of every iteration. Then, each time operator n is invoked, its success rate is updated based on the operator’s performance [43]. Success is determined when the operator either discovers a better solution or identifies a solution with the same target value but different solution structures. Recognizing that achieving optimization becomes progressively challenging with the progression of the iteration process, we factor in the rate of improvement in the objective function value when updating the success rate.

$$r_n = r_n + \begin{cases} \frac{C-C'}{C}, & \text{if } C' < C \\ \frac{1}{4C}, & \text{if } C' = C, \forall n \in N \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

4.3.4. Neighborhood Operators

Local search, a crucial component of the optimization process, is greatly enhanced by leveraging problem-specific knowledge, thereby significantly boosting the algorithm’s exploration capabilities [44,45]. The outcomes of local search are either superior novel solutions [46] or enhanced initial solutions [47]. To further enhance the algorithm’s performance, this paper introduces four tailored neighborhood structures designed for the specific problem structure to be employed during the local search process. The neighborhood structures are designed with reference to [48], as depicted in Figure 9.

Neighborhood structure InsertionInner: This structure generates neighborhood solutions by inserting an operation task into a new position. It selects an operation task on the device with the longest operation time, and then inserts it into another location on the same device. The selected operation task is chosen according to the operation time from the latest end to the earliest end, and the insertion position is selected according to the start time from the earliest start to the latest start.

Neighborhood structure InsertionBetween: This structure generates neighborhood solutions by inserting an operation task into a new position on another device of the same type. It selects an operation task on the device with the longest operation time, and then inserts it into a different location on another device of the same type. The selection of operation task and insertion position is the same as the previous neighborhood structure.

Neighborhood structure SwapInner: This structure generates new neighborhood solutions by swapping operation tasks within the same device. It selects two different operation tasks on the device with the longest operation time and swaps their positions within the same device to create a new solution. The selected operation task is selected according to the operation time from the latest end to the earliest end, and the other

operation task is selected according to the operation time from the earliest start to the latest start.

Neighborhood structure SwapBetween: This structure generates neighborhood solutions by exchanging tasks between the device with the longest operation time and another device of the same type. Specifically, it selects an operation task from the device with the longest operation time and another operation task from another device of the same type, swapping their positions to create a new solution. The choice of the two operation tasks is the same as the previous neighborhood structure.

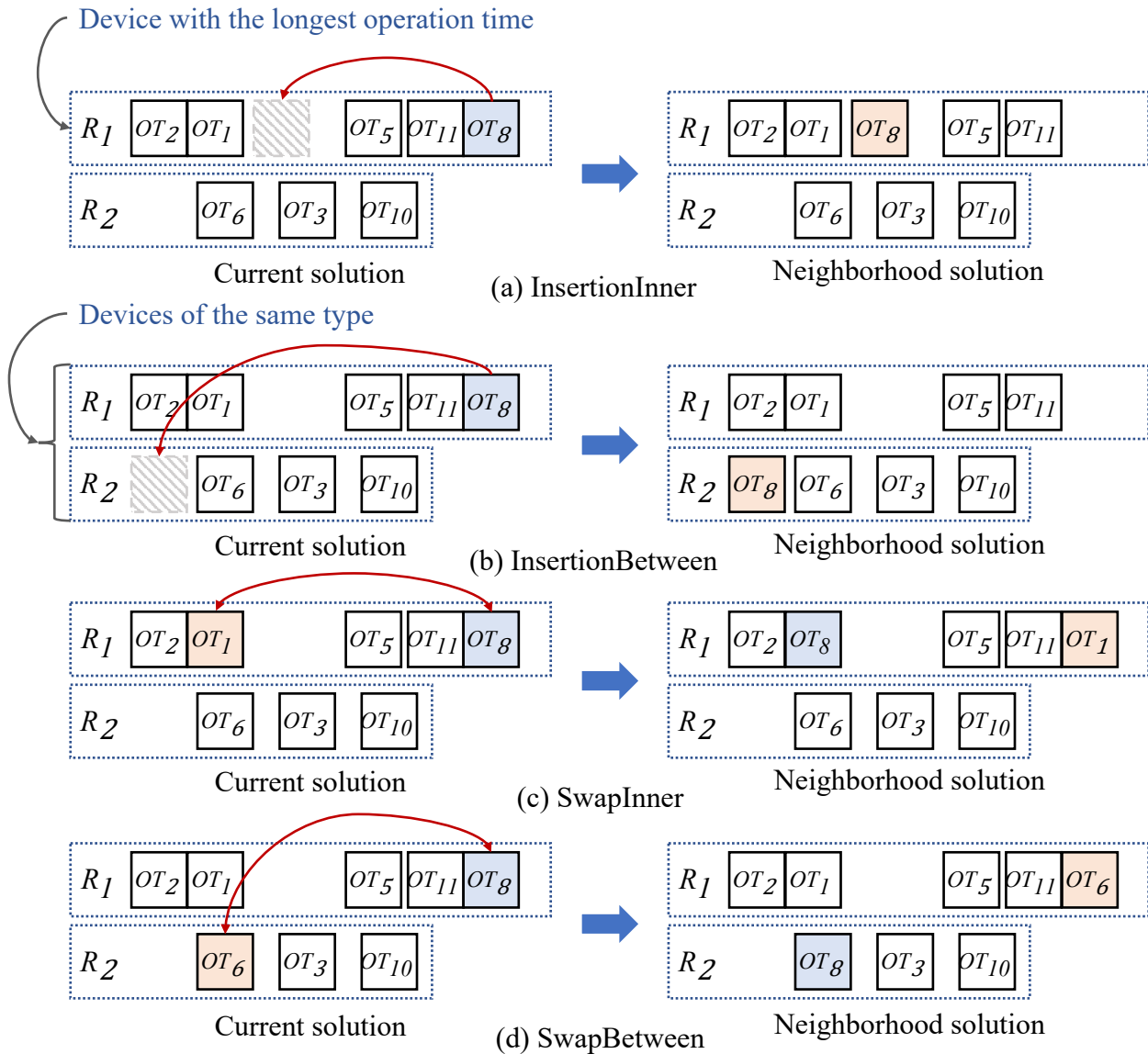


Figure 9. Illustration of neighborhood solution generation method.

4.3.5. Neighborhood Solution and Feasibility Check

The CPIIOS problem is intricate, involving a myriad of complex constraints. When examining the feasible time window for the neighborhood operators, the operators' sub-functions play a crucial role in ensuring adherence to these constraints. In this paper, the conflict refiner of the CP Optimizer with the conflict recognition function is used to identify the feasible time window on the device. On the other hand, the neighborhood operator mainly deals with constraints related to equipment operations, often neglecting crucial process-related constraints such as inbound and outbound operation precedence constraints. To address this limitation, we also use the conflict refiner to identify all constraints in the

problem and check whether the current solution is feasible. This integration allows for the identification of infeasible solutions by considering a broader spectrum of constraints, contributing to a more comprehensive and effective solution to the CPIIOS problem.

Detecting an irreducibly inconsistent set (IIS) among the constraints of a model is a standard method. However, the IIS method is only applicable to continuous linear programming models, and the conflict refiner can handle all types of continuous and discrete problems supported by CPLEX and CP Optimizer. In the CPIIOS problem, it only takes 0.009 s on average to check the feasible time window in GN5-1 using the conflict refiner, while it only takes 0.026 s on average to check the feasibility of the solution.

4.3.6. Cost Evaluation

To ensure the efficiency of the meta-heuristic algorithm, it is essential to swiftly compute the objective function value [49]. Previous approaches have often utilized linear functions to calculate processing times, employing straightforward recursive functions for objective function computation [50]. However, such calculation methods tend to sacrifice the computational time, hindering the overall efficiency of the solution process. In this study, the interval variable in the constraint programming exhibits characteristics encompassing both the start and end times. This unique feature enables the direct calculation of the objective function value by evaluating the end time of each task on respective operation devices. This approach enhances the accuracy and efficiency of the objective function calculation in the algorithm.

4.3.7. Simulated Annealing

Simulated annealing, being one of the predominant single-solution-based meta-heuristic algorithms, plays a crucial role in escaping the local optima by embracing non-improved solutions [43]. It found extensive applications in solving combinatorial optimization problems. Drawing inspiration from the threshold acceptance algorithm outlined by [51], this study employs SA to accept solutions with identical target values. In each iteration, any neighborhood solution that is at least as good as the current target value is accepted. Otherwise, the acceptance of a neighborhood solution hinges on factors including the discrepancy between the target value and the current optimal solution, along with the prevailing temperature. The Boltzmann distribution is employed to compute the acceptance probability:

$$P(\text{accept}) = e^{-\frac{s' - s}{\theta}} \quad (20)$$

4.4. Second Phase of ASALS-CP

After resolving the first phase, the most optimal feasible solution obtained is passed to the second phase, with the initial point set as the optimal solution from the first phase guiding the subsequent solver solution. The parameters and strategies for solving with the CP Optimizer in the second phase are akin to those in the first phase, as detailed in Section 4.2. Notably, this paper advocates a multi-start two-stage solution framework. Through the iterative initiation of the CP + ASALS solution framework, multiple feasible solutions are acquired, with the optimal solution chosen as the starting point for the second-stage iteration of the algorithm. The number of multiple restarts is empirically set to 20 based on parameter experiments. The operational details of the second phase of the algorithm are delineated as follows:

- The CP solver strategy in the second stage is configured similarly to the first stage, terminating when the solution time reaches the predetermined condition.
- The precision of solutions in hybrid algorithms significantly relies on the exchange of solutions between different components. To facilitate the transfer of solutions between different components, the CP Optimizer offers functionalities for setting starting points and IloSolution. By combining these features, solutions can be stored by preserving the values of decision variables, eliminating the need for encoding and decoding typically encountered in heuristic algorithms. This approach enables the

accurate transmission of solutions, enhancing the efficiency of hybrid algorithms. The concrete operational process involves reading the presence, start, and end values of key decision variables in the current feasible solution through the `IloSolution` function, assigning them to the starting point, and then transferring the solution by reading the starting point value.

5. Computational Experiments

The implementation of this paper is implemented through a Java project that combines the CP solver and ASALS. The interface to the CP solver is implemented by calling the CP Optimizer Java Archive File on the model and data files and reading its output results and logs.

Our CP solver was IBM ILOG CP Optimizer 12.10 (IBM Corporation 2020, commercial product used under academic license). Although compared to other solvers, CP Optimizer is not well known in the scheduling field. However, its simple and realistic modeling language makes it have good out-of-the-box performance for solving real-life problems [33]. The CP Optimizer's memory consumption in the tests was below 80 MB.

All experiments were conducted on a Mac Pro A2338 workstation, which was equipped with two 3.2 GHz Quad-Core Apple M1 processors, providing a total of eight cores. It had 8 GB of unified memory and 256 GB of internal storage capacity. The experiments were performed on MacOS Monterey 12.3.

5.1. Test Data

The experimental data are mainly composed of two parts. The first part is the desensitization data obtained from the actual operation data of the port under study. And the second part is the simulation data generated according to the actual operation of the coal port.

The test data in this paper are derived from the actual coal port logistics system network structure, as illustrated in Figure 1. The logistics system comprises trains, ships, stackings, and different types of equipment. Nodes representing network resources are interconnected by belts/operation streams. Detailed explanations of the data structure and generation rules for the operation streams are provided below.

To mitigate the impact of the warm-up stage (when the yard occupancy is sufficient to commence operations) on the solution results, we simulated a relevant warm-up scenario in the dataset that aligns with the actual yard operation conditions. This involves simulating an appropriate yard inventory status based on real operational conditions and subsequently excluding its influence from the solution results.

An instance of the model is mainly composed of two kinds of data: port/operation basic data and operation task information. The port/operation basic data provide the basic configuration information in the production process, as shown in Table 2. The operation task information is composed of real-life operation task data and simulation data, as shown in Table 3:

- Dataset A contains three real-life operation data entries from the port under study, and these data have been desensitized.
- Dataset B contains 16 sets of data under six data sizes and three operation scenarios. The difference between the three operation scenarios is reflected in the different stacking selections of the operation tasks. In the case of strong overlap, on average, every three operation tasks are processed on the same stacking. In the case of weak overlap, on average, every two operation tasks are processed on the same stacking. And in the case of no overlap, the operation stacking of each task is almost different.

Table 2. Basic operation data information.

Type	Value
Moving speed of equipment	30 m/min
Operation efficiency of inbound stream	300 t/min
Operation efficiency of outbound stream	400–600 t/min
Number of ship cabins	5–7
Shipment rounds	2
Inbound operation interval	50 min
Outbound operation interval	20 min
Safety distance between devices	10 m
Number of stacks	98
Optional streams for each inbound operation task	1–5
Optional streams for each outbound operation task	1–18

Table 3. Datasets and data characteristics.

Dataset	Instance/Set	Number of Inbound Streams	Number of Outbound Streams	Characteristics of Operation Scenario
Set A	R1	8	70	-
	R2	12	92	-
	R3	23	81	-
Set B	GN1	10.8	28.8	No overlap
	GW1	12	29.6	Weak overlap
	GS1	10.4	34.4	Strong overlap
	GN2	23.2	63.2	No overlap
	GW2	24	51.6	Weak overlap
	GS2	23.2	58.4	Strong overlap
	GN3	36	95.2	No overlap
	GW3	35.4	76	Weak overlap
	GS3	38	88	Strong overlap
	GN4	48.2	127.6	No overlap
	GW4	44.8	112.8	Weak overlap
	GS4	47.8	116.4	Strong overlap
	GN5	59.4	148.4	No overlap
	GW5	58.2	147.6	Weak overlap
GS5	59.2	137.2	Strong overlap	
GN6	60	177	No overlap	

5.2. Algorithm Parameter Settings

The parameter configuration and solution strategy of the CP Optimizer have been extensively discussed in detail in Section 4.2. This section primarily delves into the setup of the ASALS algorithm to optimize its performance, prevent premature convergence, and enhance the diversity of the algorithm’s search space. The ASALS algorithm proposed in this paper encompasses five key parameters: the maximum number of iterations $MaxIter$, the iteration optimization time for operators τ , the reaction factor α , the cooling rate μ , and the minimum selection probability p^{min} . Each parameter consists of five different levels, such as $MaxIter = [10, 30, 50, 70, 90]$, $\tau = [0.1, 0.2, .0.3, 0.4, 0.5]$, $\alpha = [0.1, 0.2, 0.3, 0.4, 0.5]$, $\mu = [0.90, 0.92, 0.94, 0.96, 0.98]$, $p^{min} = [0.01, 0.02, 0.03, 0.04, 0.05]$. For the sake of ensuring robustness and stability, all parameter combinations are independently executed 10 times under a predefined termination condition (i.e., 600 s), and the average objective function value is used as the metric. Figure 10 illustrates the trend of factor levels for the five parameters. Based on the figure, the ASALS algorithm parameters in this paper are configured as follows: $MaxIter = 50$, $\tau = 0.4$, $\alpha = 0.5$, $\mu = 0.9$, $p^{min} = 0.02$.

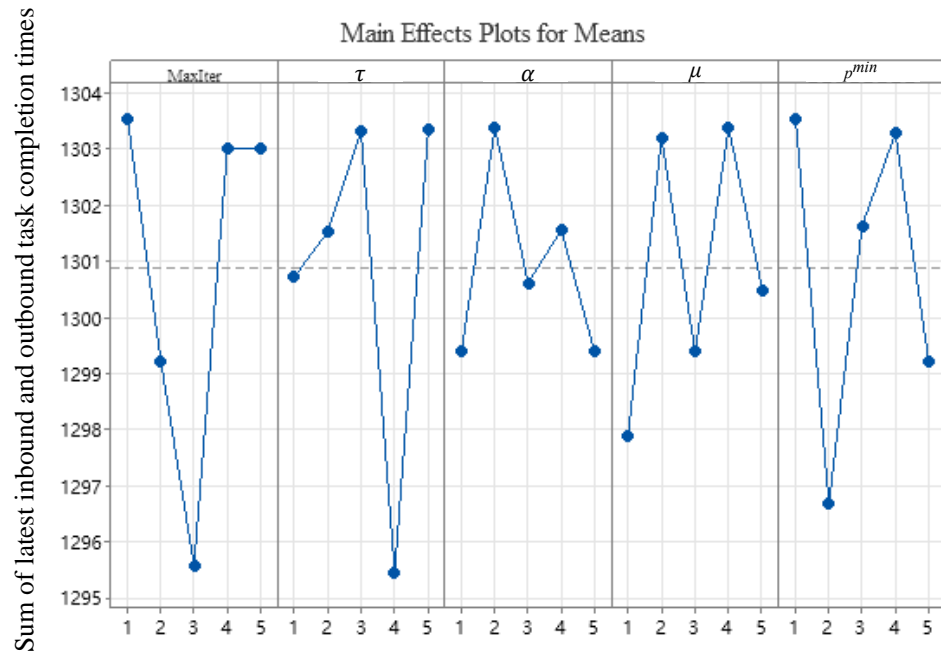


Figure 10. Factor level trend of five key parameters.

5.3. Performance Comparison with Other Algorithms

To assess the performance of the CP method and the proposed ASALS algorithm, this study compares them with two widely used VNS algorithm [52] and SA algorithm [53] applied to port scheduling problems. The parameter settings for these two algorithms are obtained from [54,55], as detailed in Table 4. In this context, T_{Start} and T_{end} denote the initial and ultimate temperatures in the SA, and L signifies the number of iterations at each temperature in the SA algorithm. Given the substantial influence of the initial solution on algorithmic efficiency, this study leverages the CP solver to construct initial solutions for both algorithms, which are subsequently subjected to optimization processes using the respective algorithms. Considering the inherent stochastic nature of heuristic algorithms, each algorithm undergoes 10 trials in a consistent environment. Table 5 presents the average objective values and average GAP values, with the best results for each combination highlighted in bold. The GAP value represents the percentage of the relative difference between the current objective value and the lower bound given by the CP Optimizer. Note that the lower bound provided by the CP Optimizer may be obtained after relaxing some operational constraints, resulting in a smaller GAP value between the objective value and the actual optimal value.

Table 4. The parameters for the compared algorithms.

Algorithms	Parameters
VNS	$MaxIter = 10$
SA	$T_{Start} = 2000, T_{end} = 0.001, \mu = 0.95, L = 100$

According to the information provided in Table 5, the following conclusions can be drawn.

Table 5. Comparison of CP Optimizer, SA, VNS, and ASALS algorithms.

Instance	CP Optimizer		CP Optimizer (Solution Time Is 1000 s)		CP-SA		CP-VNS		ASALS-CP	
	Time (mins)	GAP(%)	Time (mins)	GAP(%)	Time (mins)	GAP(%)	Time (mins)	GAP(%)	Time (mins)	GAP (%)
GN1-1	607	10.21	607	10.21	607	10.21	607	10.21	607	10.21
GN1-2	739	0	739	0	739	10.83	739	10.83	739	0
GN1-3	735	0	735	0	735	6.80	735	6.80	735	0
GN1-4	478	12.55	478	12.55	478	12.55	478	12.55	478	12.55
GN1-5	386	0	386	0	386	0	386	7.77	386	0
GN2-1	711	5.91	711	5.91	711	5.91	711	5.91	711	5.91
GN2-2	1125	22.93	1125	22.93	1125	22.93	1125	22.93	1198	21.04
GN2-3	796	1.88	796	1.88	796	1.88	796	1.88	796	0
GN2-4	755	3.71	755	3.71	755	3.71	755	0	755	0
GN2-5	635	9.45	635	9.45	635	9.45	635	9.45	635	9.45
GN3-1	782	9.85	782	9.85	782	9.85	782	9.85	782	9.85
GN3-2	976	1.54	976	1.54	976	0	976	0	976	0
GN3-3	1026	15.89	1026	15.89	1026	15.89	1026	15.89	1025	15.80
GN3-4	1051	17.98	1051	17.98	1049	17.83	1049	17.83	1049	17.83
GN3-5	1111	20.16	1111	20.16	1111	20.16	1110	20.09	1104	19.66
GN4-1	985	12.08	985	12.08	985	12.08	985	12.08	985	12.08
GN4-2	986	12.98	986	12.98	986	12.98	986	12.98	985.2	12.89
GN4-3	1182	15.31	1182	15.31	1184	15.46	1179.6	15.10	1177	14.95
GN4-4	1205	16.93	1205	16.93	1247	19.73	1198	16.44	1181	15.24
GN4-5	1212	17.41	1212	17.41	1212	17.41	1214	17.55	1185	15.53
GN5-1	1303	15.27	1303	15.27	1284	14.02	1303	15.27	1284	14.02
GN5-2	1277	14.57	1277	14.57	1277	14.57	1281.4	14.84	1273	14.30
GN5-3	1218	23.40	1218	23.40	1218	23.40	1218	23.40	1213.8	23.13
GN5-4	1303	16.27	1303	16.27	1306	16.46	1277.4	14.59	1271	14.16
GN5-5	1323	16.55	1323	16.55	1323	16.55	1323	16.55	1302	15.21

Note: the notation GN*i*-*j* denotes the *j*th instance generated in accordance with the generation rules of the dataset GN*i*. For instance, GN1-1 represents the first instance generated based on the generation rules of the GN1 dataset.

- The CP Optimizer demonstrated a consistent performance across all instances, providing feasible solutions with GAP values below 30%, although many instances reach the solution time limit. In some cases, it is a struggle to significantly improve the solution results even with the increased solution time (CP (1000 s)). Instances achieved optimal solutions in 12% of cases, and 88% of instances showed a GAP of less than 20%. The solver’s efficiency was notable, especially in scenarios where optimal solutions were readily attainable. Across the GN1–GN5 sets, the CP optimizer exhibits average solution gaps of 4.55%, 8.78%, 13.08%, 14.94%, and 17.21%, respectively. This indicates the CP Optimizer’s capability to quickly attain near-optimal solutions within a defined timeframe. However, its effectiveness diminishes with larger problem scales, as evidenced by increased task completion times. The detailed introduction of different data scales and difficulty is shown in Section 5.9.
- The ASALS algorithm demonstrates notable improvements, achieving a substantial enhancement in the solution results, especially in larger and more challenging instances. The two-stage algorithm combining ASALS and CP solver showed a notable improvement of up to 6.49% in the best-case scenario. This showcases its ability to efficiently explore the search space and avoid local optima. In certain scenarios, notably for small-scale or easily optimized instances, further optimization proves challenging for ASALS-CP. Compared with the CP solver, ASALS outperforms the CP solver in 17 of 25 instances. Specifically, the average solution gaps obtained by ASALS for GN1–GN5 are 4.55%, 7.28%, 12.63%, 14.14%, and 16.16%, respectively, representing a gradual increase from 0 to 6.49% compared to the average improvement in the CP solver. This trend indicates that, as the problem scale expands, the advantages of the ASALS algorithm become increasingly evident. Nevertheless, its robustness is evident in handling instances of diverse sizes and complexities, providing alternative solutions while maintaining a balance between solution quality and computational efficiency.
- Compared with SA and VNS algorithms, the ASALS algorithm achieves a good balance between convergence speed and traversing the search space. It outperforms these algorithms in terms of both efficiency and the ability to avoid falling into local optima. The ASALS algorithm outperforms the SA and VNS algorithms in 15 out of 25 and 16 out of 25 instances, respectively. On average, the ASALS algorithm

consistently achieves better objective function values and gaps compared to the SA and VNS algorithms. Figure 11 illustrates the convergence of objective values for the four algorithms across different instances. In comparison to the rapid convergence and susceptibility to local optima of the CP solver, as well as the slow convergence and difficulty in escaping local optima exhibited by VNS and SA algorithms, the ASALS algorithm strikes a commendable balance between convergence speed and thorough exploration of the search space.

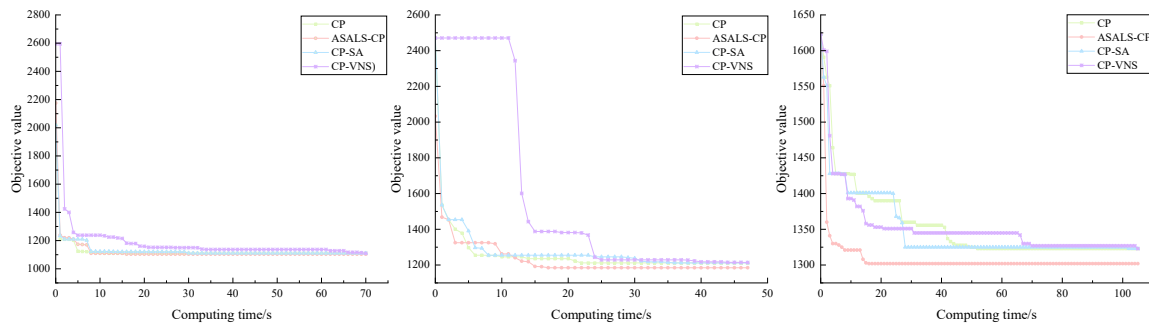


Figure 11. The convergence process of objective values of different algorithms under different instances. From left to right is the convergence process of objective values of GN3-5, GN4-5, and GN5-5.

Figure 12 presents a comparative analysis of scheduling outcomes between the CP solver and the ASALS-CP algorithm. The results underscore the efficiency of the ASALS-CP algorithm. Through the optimized arrangement of equipment operation tasks, the ASALS-CP algorithm achieves a more time-efficient scheme, enhancing operational efficiency in the coal port production process. It is noteworthy, however, that the operation scheme derived from the two-stage algorithm notably differs from that of the CP solver, not only in terms of operation time but also in the selection of operation stacking. This divergence implies that the CP Optimizer may encounter challenges escaping local optima during the solution process, limiting its ability to explore solutions with significantly improved objective values.

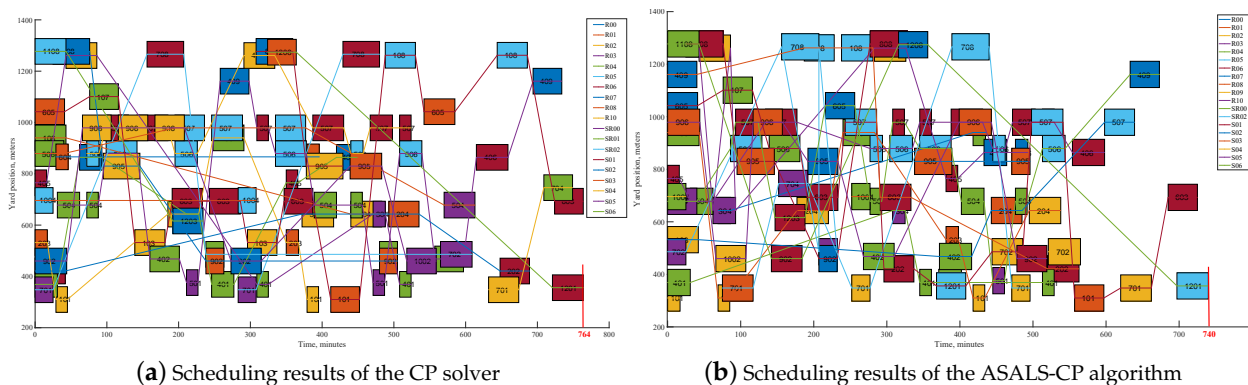


Figure 12. Illustration of the effect of the CP solver (a) versus the ASALS-CP algorithm (b) on the scheduling results of instance GN5-1.

To ascertain the adaptability of the ASALS algorithm in handling extreme port operations and to demonstrate its efficacy in large-scale scenarios, this study compares and analyzes the solution efficiency of the ASALS algorithm and the CP algorithm using a set of large-scale instances, as depicted in Table 6. The findings indicate that the ASALS algorithm not only reduces the operation time and solution gap but also enhances the yard and equipment occupancy rates for large-scale instances. This occurs because the ASALS algorithm aims to optimize the objective function by minimizing the operation time, leading it to prioritize parallel operations. This strategy involves selecting more operation equipment

and operation stacking, as evidenced by the comparative analysis of the above solution schemes. These results underscore the robust performance of the proposed algorithm in busy port operations and underscore its effectiveness and efficiency compared to the direct solution method of the CP solver, highlighting its strong practical applicability.

Table 6. Comparison of the results of the CP solver and the ASALS-CP algorithm for challenging instances.

Method	Index	GN6-1	GN6-2	GN6-3
CP Optimizer	Time (mins)	1194	1192	1168
	GAP (%)	21.86	21.73	20.12
	Yard occupancy rate (%)	71.13	73.19	76.29
	Equipment occupancy rate (%)	68.42	73.68	73.68
ASALS-CP algorithm	Time (mins)	1186	1176	1160
	GAP (%)	21.33	20.66	19.57
	Yard occupancy rate (%)	71.13	75.26	76.29
	Equipment occupancy rate (%)	78.95	78.95	84.21

5.4. Comparison with Manual Scheduling Method

To verify the feasibility of our solutions, we compare the manual scheduling results of three actual instances with the scheduling results obtained by the two-stage algorithm. Table 7 demonstrates that the ASALS-CP algorithm can achieve optimal scheduling results within a short solution time (not exceeding 15 s), whereas manual scheduling typically requires more than 20 min to generate a feasible scheduling scheme.

The relative difference percentage quantifies the variation in operation completion times between the two scheduling approaches: the ASALS-CP algorithm and manual scheduling. The scheduling scheme obtained by the ASALS-CP algorithm significantly reduces the task completion time, with the maximum operation time shortened by nearly 60% compared to manual scheduling. However, it is important to note that, as the problem scale increases, this advantage may experience a certain decline. Meanwhile, the ASALS-CP algorithm effectively enhances the yard and equipment occupancy rates, improving the yard occupancy rate by 12.03% and the equipment occupancy rate by 8.33%. It is worth noting that the yard occupancy rate at the studied port is relatively low due to strategic reserves, with only about 50% of yard resources being utilized for actual operations.

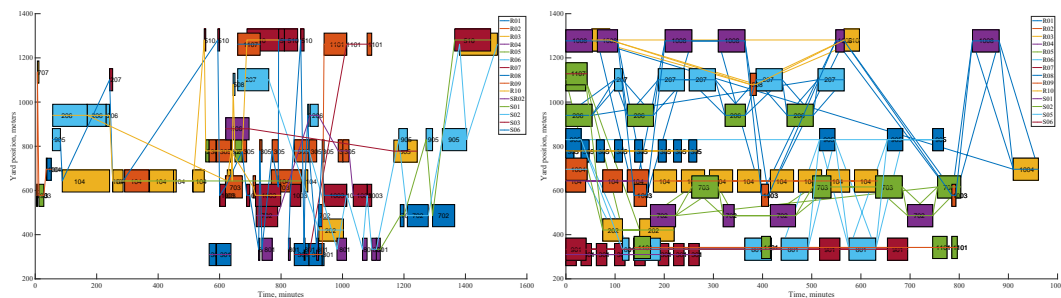
Figure 13 presents the scheduling scheme diagrams obtained by the manual scheduling and the ASALS-CP algorithm for solving the instance of R1. It is evident that the manual scheduling scheme is more loosely arranged with lower equipment utilization rates, whereas the scheduling scheme generated by the ASALS-CP algorithm is significantly more compact, resulting in shorter task completion times.

Table 7. Comparison of the results of the manual scheduling method and the ASALS-CP algorithm.

Method	Index	R1	R2	R3
Manual scheduling method	Inbound operation volume (tons)	33,412	52,246	95,187
	Inbound operation completion time (mins)	1047	916	1448
	Outbound operation volume (tons)	222,103	236,614	272,539
	Outbound operation completion time (mins)	2640	1311	1264
	Yard occupancy rate (%)	17.53	25.77	27.55
	Equipment occupancy rate (%)	70.59	89.47	63.16
ASALS-CP algorithm	Inbound operation volume (tons)	34,255	51,993	97,823
	Inbound operation completion time (mins)	457	402	975
	Outbound operation volume (tons)	213,549	237,595	253,785
	Outbound operation completion time (mins)	1088	1128	1140
	Yard occupancy rate (%)	18.55	28.87	27.84
	Equipment occupancy rate (%)	73.68	94.74	68.42
-	Relative difference (%)	58.79	13.96	9.81

The comparison of scheduling results reveals that the ASALS-CP algorithm achieves more efficient operation equipment and stacking utilization, with more compact operation tasks. In contrast, manual scheduling often involves frequent equipment switches within a short period and larger distances between the adjacent operation stacks. These factors contribute to the significantly longer operation time observed in manual scheduling compared to the ASALS-CP algorithm.

These results demonstrate the feasibility and effectiveness of our proposed ASALS-CP algorithm for the CPIIOS problem. The algorithm not only achieves optimal scheduling results within a short solution time but also improves the task completion efficiency compared to manual scheduling, especially for smaller-scale instances.



(a) Scheduling scheme diagram of the manual scheduling method (b) Scheduling scheme diagram of the ASALS algorithm

Figure 13. Scheduling scheme diagrams of R1 obtained by the manual scheduling method (a) and ASALS algorithm (b).

5.5. Comparison to a Genetic Algorithm without CP Solver

The genetic algorithm has proven to be an effective and versatile approach for solving combinatorial optimization problems, particularly resource-constrained scheduling problems [56]. Zhang et al. proposed a heuristic algorithm based on NSGA-II and VNS to address the combination problem of loading operation planning and ship traffic scheduling in bulk ports [15]. Building upon Zhang et al.’s work, this paper devises a genetic algorithm tailored to the CPIIOS problem.

In the designed genetic algorithm, a chromosome is defined, comprising segments for inbound operation tasks and outbound operation tasks. Feasible solutions are generated through random generation, and the algorithm incorporates standard genetic steps such as selection, crossover, mutation, and retention. However, experimental results demonstrate that finding solutions that adhere to all constraints is challenging for the genetic algorithm. Even for the smallest-scale example GN1-1, the genetic algorithm can only attain a solution with an operation completion time of 697 when disregarding the no-overlap constraints on operation equipment and operation stacking, which is still close to 15% worse than the solution obtained by the CP Optimizer. This limitation highlights the difficulty of achieving feasible solutions for complex combinatorial optimization problems with intricate constraints, as presented in this study. To enhance the efficiency of such algorithms, future endeavors may focus on improving the quality of initial solutions and incorporating repair operators to address constraint violations.

5.6. Scheduling Results and Equipment Operation Diagrams

Figure 14 illustrates the scheduling scheme diagrams obtained by the ASALS-CP algorithm for five instances with progressively larger task scales. As the scale of tasks increases from diagram (a) to diagram (e), the corresponding operation completion times and the number of operation equipment in the diagrams also increase. Notably, the diagrams reveal a noticeable trend towards a more compact arrangement of tasks as the scale increases. This demonstrates the algorithm’s ability to effectively handle larger task sets while optimizing the overall scheduling scheme.

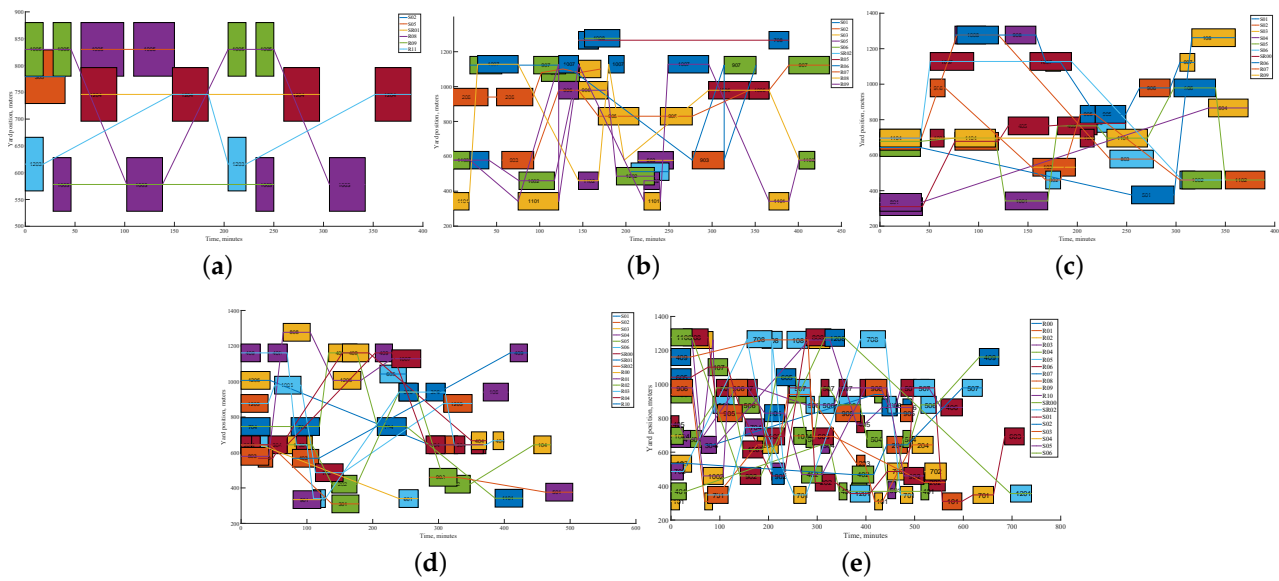


Figure 14. Scheduling scheme diagrams obtained by the solutions of five instances. The results for GN1-1, GN2-1, GN3-1, GN4-1, and GN5-1 are shown from (a) to (e).

The diagrams vividly illustrate the impact of task scale on equipment operations. As the scale of tasks increases, the demand for completing tasks on each equipment intensifies. This heightened task allocation poses significant challenges in ensuring the uniqueness constraint of each piece of equipment and complying with the constraints related to the equipment’s movement distance. Nevertheless, the ASALS-CP algorithm proves its effectiveness in efficiently managing these complexities and delivering optimized scheduling solutions for equipment operations. The diagrams serve as visual evidence of the algorithm’s capability to handle diverse task scales and generate well-organized equipment operations.

5.7. Choice of Objective Function

The coal port integrated inbound and outbound scheduling system is a discrete-time dynamic system, and establishing a direct correlation between various performance indices and system performance is challenging. Furthermore, scheduling optimization results can significantly differ based on the same indicator, considering the diversity of coal port systems and customer needs. To enhance the universality and practicality of the coal port integrated scheduling system, it is crucial to optimize the system performance using various scheduling indicators, thereby offering a diverse set of scheduling strategies.

- Minimizing the operation completion time: one objective to improve the coal port’s efficiency and service level is to minimize the completion time of the operations.
- Maximizing the average equipment utilization: This optimization aims to enhance the efficiency of equipment operation and optimize resource allocation. This is accomplished by maximizing the utilization rate, calculated as the percentage of the equipment operation time to the latest completion time of all operations.
- Minimizing equipment load imbalance: The objective is to minimize the equipment load discrepancies and eliminate the efficiency losses caused by operational imbalances, which is achieved by calculating the variance of the equipment operating time.

In Table 8, we analyze the performance of the three objective functions mentioned above using instances R1–R3, with the default objective function being the minimization of the operation completion time. It is important to note that all data in the table have been desensitized to maintain confidentiality. It is evident that different objective functions yield advantages in different evaluation indicators. However, the objective function employed in this paper consistently performs well across all aspects of the indicators.

Table 8. Objective function versus value: operation completion time, average equipment utilization, and equipment load imbalance.

Instance	Objective Functions	Time (mins)	Utilization (%)	Imbalance (mins ²)
R1	Operation completion time	1276	31.32	87892.75
	Average equipment utilization	1427	32.87	44578.75
	Equipment load imbalance	2269	19.13	224708.10
R2	Operation completion time	1362	30.72	120592.10
	Average equipment utilization	1497	30.19	87892.75
	Equipment load imbalance	1420	32.63	100948.45
R3	Operation completion time	1066	46.07	115431.15
	Average equipment utilization	1249	47.80	120173.15
	Equipment load imbalance	1109	43.42	109858.49

5.8. Model Sensitivity Analysis under Different Model Parameters

Table 9 shows operation completion time for the five instances under different model parameters, using CP Optimizer as the backend without ASALS. And Figure 15 compares the impacts of different model parameters on the operation task completion time and GAP. The results reveal that, in comparison to the inbound operation, the outbound operation significantly influences the difficulty of solving the model. Neglecting the inbound operation leads to a considerable increase in the complexity of solving the CPIOOS problem. The remaining operational constraints also exert a certain influence on the model’s difficulty. Among these, the no-overlap constraint, which ensures operation equipment and stockpile uniqueness, has the most significant impact. By removing this constraint, an optimal solution can be obtained for any problem scale. Conversely, the inbound operation precedence constraint has the least impact on the model’s difficulty.

Table 9. The influence of model parameters on operation task completion time and GAP value.

	GN1-1		GN2-1		GN3-1		GN4-1		GN5-1	
	Time (mins)	GAP (%)	Time (mins)	GAP (%)	Time (mins)	GAP (%)	Time (mins)	GAP (%)	Time (mins)	GAP (%)
Default: CP Optimizer	607	10.21	711	5.91	782	9.85	985	12.08	1303	15.27
ASALS-CP	607	10.21	711	5.91	782	9.85	985	12.08	1284	14.02
No inbound operations	342	0	436	9.63	392	19.39	465	22.15	443	32.28
No outbound operations	218	0	275	0	390	0.26	504	0	531	0
No stacking capacity limit	218	0	275	0	390	0.26	504	0	531	0
No inbound precedence limit	194	0	208	3.85	354	1.69	354	1.41	400	0
No outbound precedence limit	218	0	275	0	390	0.26	504	0	531	0
No coal blending limit	218	0	275	0	390	0.26	504	0	531	0
No no-overlap limit	195	0	275	0	356	0	504	0	502	0
No device moving distance limit	218	0	275	0	372	4.3	504	0	527	0
No device crossing limit	218	0	275	0	390	0.26	504	0	531	0

This underscores the meaningfulness of considering the integrated scheduling problem of inbound and outbound operations in this study. And each constraint in the model carries significance for problem-solving. Ignoring a specific operation aspect or constraint introduces varying levels of difficulty in solving the problem.

5.9. Model Sensitivity Analysis by Multi-Data Scale

Table 10 presents the time taken to find the optimal solution for the 75 instances and the corresponding GAP under various data scales and operation scenarios, utilizing CP Optimizer as the backend without ASALS. Figure 16 visually depicts the influence of data size and scenarios on the time taken to find the optimal solution and the GAP. The figure illustrates that, as the data scale increases, the GAP value also increases, indicating a decline in the solution quality. This implies that larger data scales result in relatively worse solution outcomes. Additionally, more complex operation scenarios have a greater negative impact on solution quality compared to the data scale alone. Furthermore, the increase in data scale and complexity leads to longer times required to find the optimal solution.

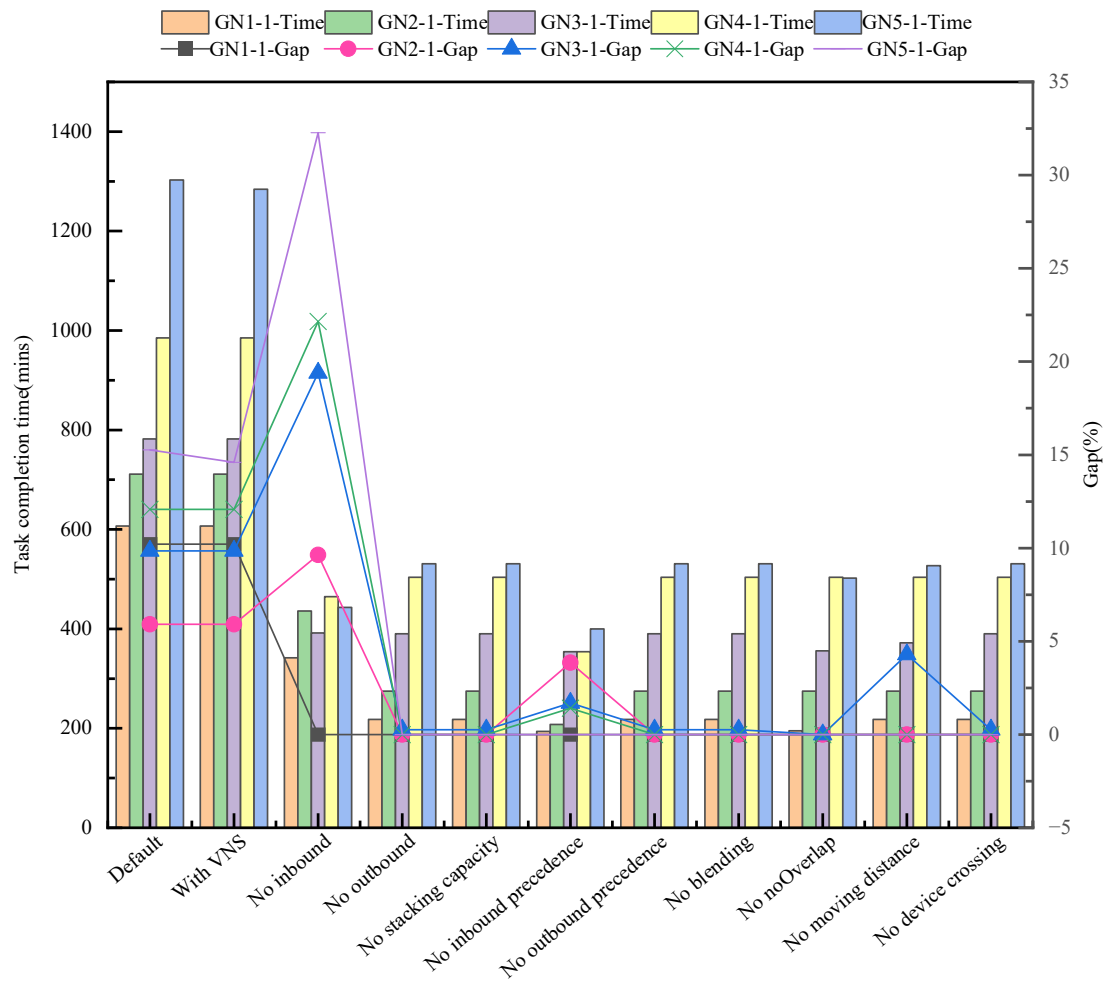


Figure 15. Comparison of the operation task completion time and GAP under different model parameters.

However, as the number of operation tasks increases, the complexity of the solving scenario can actually expedite the solver’s ability to find the optimal solution. This phenomenon may be attributed to the complexity of the scenarios and constraints, which make it easier for the constraint propagation solver to narrow down the solution space and quickly identify feasible solutions.

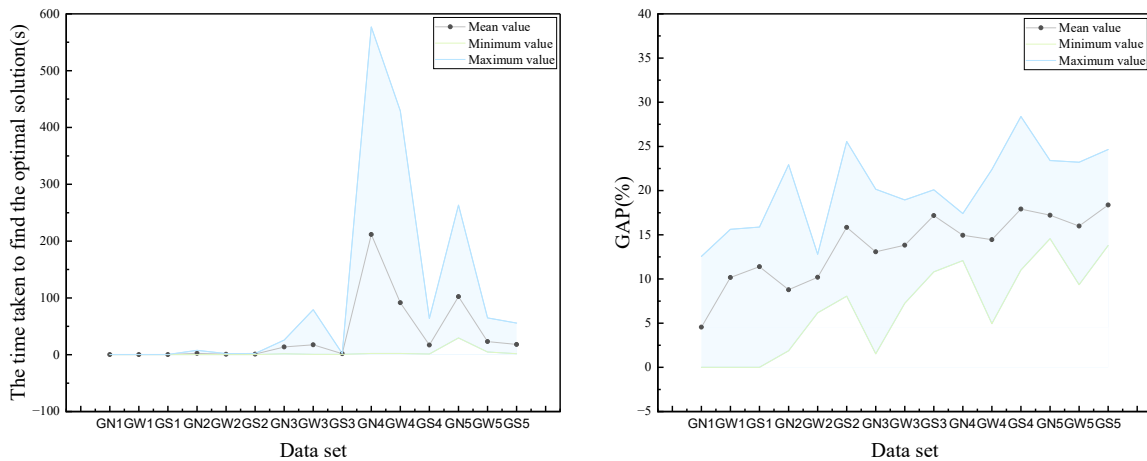


Figure 16. Impact of data sizes and scenarios on the time taken to find the optimal solution (s) (left) and GAP (%) to the optimal solution (right).

Table 10. Model sensitivity analysis by multi-data scale and operation scenario.

Instance	The Time Taken to Find the Optimal Solution (s)	GAP (%)
GN1-1	0.31	10.21
GN1-2	0.35	0
GN1-3	0.25	0
GN1-4	0.37	12.55
GN1-5	0.34	0
GW1-1	0.23	15.62
GW1-2	0.22	0
GW1-3	0.29	8.84
GW1-4	0.46	12.09
GW1-5	0.25	14.32
GS1-1	0.31	14.67
GS1-2	0.25	12.50
GS1-3	0.29	0
GS1-4	0.25	15.88
GS1-5	0.38	13.90
GN2-1	1.99	5.91
GN2-2	7.75	22.93
GN2-3	1.30	1.88
GN2-4	0.61	3.71
GN2-5	0.23	9.45
GW2-1	2.01	12.80
GW2-2	0.47	10.88
GW2-3	0.25	6.17
GW2-4	0.66	12.20
GW2-5	0.64	8.91
GS2-1	0.29	8.06
GS2-2	1.74	25.56
GS2-3	2.15	22.86
GS2-4	1.18	12.12
GS2-5	0.30	10.56
GN3-1	1.57	9.85
GN3-2	7.81	1.54
GN3-3	11.75	15.89
GN3-4	21.38	17.98
GN3-5	25.84	20.16
GW3-1	0.75	8.00
GW3-2	1.37	7.25
GW3-3	1.09	16.29
GW3-4	5.22	18.66
GW3-5	79.29	18.94
GS3-1	0.61	17.30
GS3-2	2.15	18.17
GS3-3	1.88	10.81
GS3-4	2.22	20.09
GS3-5	2.08	19.50
GN4-1	2.2	12.08
GN4-2	154.21	12.98
GN4-3	279.75	15.31
GN4-4	577	16.93
GN4-5	45.24	17.41
GW4-1	16.88	9.70
GW4-2	2.58	16.94
GW4-3	7.42	18.30
GW4-4	2.14	4.95
GW4-5	429.20	22.38
GS4-1	13.11	28.38
GS4-2	2.02	16.29
GS4-3	1.15	11.01
GS4-4	4.86	12.59
GS4-5	63.86	21.28
GN5-1	29.52	15.27
GN5-2	55.25	14.57
GN5-3	82.29	23.40
GN5-4	263.45	16.27
GN5-5	80.50	16.55
GW5-1	10.51	21.28
GW5-2	64.75	23.21
GW5-3	11.44	9.36
GW5-4	4.73	10.72
GW5-5	37.28	14.91
GS5-1	11.17	16.46
GS5-2	7.58	13.79
GS5-3	55.78	17.30
GS5-4	1.78	17.77
GS5-5	7.01	24.65

6. Conclusions

We employ ASALS and CP techniques to tackle a complex coal port integrated inbound and outbound scheduling problem. Our objective is to develop a reliable and efficient operational scheduling scheme for medium- and short-term operations at coal ports. By utilizing the simple and realistic characteristics of the constraint programming modeling language, we can effectively and appropriately model the operational processes of the coal port. Given the port layout, production process, and operational constraints of the studied port, we constructed an integrated scheduling model for the actual production operations of composite coal ports. This model considers the port inbound, storage, and outbound operations, as well as the coal blending process based on ship demand. The integrated scheduling of inbound and outbound operations is found to have a significant influence on the model's solution, directly impacting the overall difficulty of solving the problem. Additionally, we observe the indispensability of the various operational constraints proposed in this study. Disregarding any of these constraints affects the problem's structure and changes the complexity of finding the solution.

Recognizing the limitations of constraint programming in solving large-scale problems, as it is prone to obtain trapped in local optimum, we propose a two-stage solution method that combines CP and ASALS to enhance the solution process for the CPIIOS problem. The customized search capabilities of the CP solver allow for resource allocation with a greedy approach, which may lead to a local optimum. Employing the CP Optimizer, a high-quality initial solution is obtained, followed by the application of an ALS algorithm for local optimization. The SA algorithm is introduced to accept diverse scheduling schemes with different solutions under the same objective function, facilitating the escape from local optima and further enhancing the solution efficiency. It is important to highlight that the ASALS algorithm proposed in this paper consistently outperforms or matches the performance of the CP solver, the SA, and the VNS algorithms across all instances. By employing this improved algorithm, a 7.26% improvement in the GAP can be achieved. The CP method proposed in this paper requires significant computational time to escape local optima. In contrast, the ASALS algorithm can produce superior solutions in a shorter time, especially for large-scale instances. This capability enables more frequent and effective adjustments to the operational plan in the dynamic scheduling environment of large coal ports, allowing them to adapt to dynamic environmental changes such as equipment failures, and variations in train and ship arrival times.

Comparing the results with manually generated schedules, we find that the constraint programming-based results align with the actual operational requirements of the coal port. Moreover, the proposed approach significantly improves both the solution efficiency and solution quality compared to manual planning and scheduling. Out of the 75 instances, we achieve optimal solutions in 8 cases, and for all instances, the solution gap is below 30%. Consequently, we have confidence in the ability of the proposed method to obtain high-quality near-optimal solutions. This method offers flexibility and scalability advantages: it employs a straightforward constraint programming modeling approach that closely reflects the actual operational context, allowing port managers to modify or expand the model based on the port's production and operational processes. However, its drawback lies in the lack of a flexible and efficient search strategy, which occasionally hinders one's efforts to find the optimal solution.

Future research endeavors should center around more intricate coal port integrated planning and scheduling models, particularly incorporating ship scheduling, berth allocation, and dumper allocation. The development of models encompassing both inbound and outbound operation scheduling would be crucial for aiding port managers in implementing comprehensive and efficient port management practices. Moreover, the effectiveness of our proposed method in handling large-scale problems still requires enhancement. Exploring and devising more efficient and flexible solving tools can further improve the solving time and effectiveness. Researchers can explore the effects of various port management strategies on operational efficiency and problem structure. For instance, they might examine the

impact of merging the demand orders from different ships for coal blending operations, enhancing the overall port efficiency by efficiently transporting the same product coal to multiple ships. Given the increasing consensus within the port industry on addressing the rising energy prices and mitigating climate change [57], researchers can also delve into studies on port carbon emissions and energy efficiency. Such research endeavors can contribute to promoting energy conservation and emission reduction in ports, thus fostering their sustainable and environmentally friendly development.

Author Contributions: Conceptualization, X.L., Y.Z. and L.Z.; investigation, X.L. and L.Z.; methodology, X.L., Y.Z. and L.Z.; writing—original draft preparation, X.L., Y.Z. and L.Z.; writing—review and editing, X.L., Y.Z., L.Z., C.Y. and J.W.; supervision, Y.Z. and L.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Grant No. 72174160), the Key R & D projects in Hubei Province (Grant No. 2023BAB073), and the Fundamental Research Funds for the Central Universities (Grant No. 2023-vb-011).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available due to limitations of the study phase.

Conflicts of Interest: Author Y.Z. was employed by the company Guangdong Inland Port and Shipping Industry Research Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Parameters

T	Time set in minutes when each operation stream can start the operation in the scheduling period, $[T] = \{1, 2, \dots, T \}$
H	Set of inbound operation tasks in the scheduling period
V	Set of outbound operation tasks in the scheduling period
B	Set of coal blending schemes
M	Set of all production resources on the stockyard, including dumpers, belt lines, stackers, reclaimers, shiploaders, etc.
R	Set of stackers and reclaimers
S	Set of stockpiles
G	Set of yard equipment tracks
$s(h)$	Operation stockpile of inbound operation task h , $s(h) \in S$
$s(v)$	Operation stockpile of outbound operation task v , $s(v) \in S$
$F(h)$	Set of optional operation streams of inbound operation tasks h
$F(v)$	Set of optional operation streams of outbound operation tasks v
$R(g)$	Set of equipment sharing track g , $R(g) \in R$
siz_h^{unload}	Operation volume of inbound operation task h
siz_v^{load}	Operation volume of outbound operation task v
cap_f^{unload}	Operation efficiency of inbound operation stream f (t/min)
cap_f^{load}	Operation efficiency of outbound operation stream f (t/min)
ord_h^{unload}	Operation sequence of inbound operation task h in the dumper operation set
ord_v^{load}	Operation sequence of outbound operation task v in the ship operation set
$lead$	Operation task switching time of equipment
est_h^{unload}	The earliest time when inbound the operation task h can start operation
	$(est_h^{unload} = est_{h'}^{unload} + \frac{siz_{h'}^{unload}}{cap_f^{unload}} + lead, ord_h^{unload} = ord_{h'}^{unload} + 1)$

est_v^{load}	The earliest time when outbound operation task v can start operation ($est_v^{load} = est_v^{load} + \frac{siz_v^{load}}{cap_f^{load}} + lead, ord_v^{load} = ord_v^{load} + 1$)
inv_0^s	Initial inventory tonnage of stockpile s
cap_s	Maximum inventory of stockpile s
cap_m	Total quantity of available resources of the same type as m
pos_s	Distance from the middle point of stockpile s to the starting point of strip yard
pos_0^r	Initial location of equipment r
spd_r	Movement speed of equipment r (m/min)
dis	Safety distance between pieces of equipment on the same track
Variables	
f	Operation stream selected to perform inbound or outbound operation task
i_t^s	Inventory tonnage of stockpile s at time t
i_t^m	The amount of resource m occupied at time t
p_t^r	Location of equipment r at time t
$F(s)^{unload}$	Set of all inbound operation streams on stockpile s
$F(s)^{load}$	Set of all outbound operation streams on stockpile s
$F(m)$	Set of all operation streams on production resource m

References

- United Nations Conference on Trade and Development. *Review of Maritime Transport 2022*; United Nations Office: Geneva, Switzerland, 2022; pp. 1–23.
- Boland, N.; Gulczynski, D.; Savelsbergh, M. A Stockyard Planning Problem. *EURO J. Transp. Logist.* **2012**, *1*, 197–236. [\[CrossRef\]](#)
- Boland, N.; Kalinowski, T.; Waterer, H.; Zheng, L. Mixed Integer Programming Based Maintenance Scheduling for the Hunter Valley Coal Chain. *J. Sched.* **2013**, *16*, 649–659. [\[CrossRef\]](#)
- Boland, N.; Kalinowski, T.; Waterer, H.; Zheng, L. Scheduling Arc Maintenance Jobs in a Network to Maximize Total Flow over Time. *Discret. Appl. Math.* **2014**, *163*, 34–52. [\[CrossRef\]](#)
- Babu, S.A.K.I.; Pratap, S.; Lahoti, G.; Fernandes, K.J.; Tiwari, M.K.; Mount, M.; Xiong, Y. Minimizing Delay of Ships in Bulk Terminals by Simultaneous Ship Scheduling, Stockyard Planning and Train Scheduling. *Marit. Econ. Logist.* **2015**, *17*, 464–492. [\[CrossRef\]](#)
- Pratap, S.; Kumar B, M.; Saxena, D.; Tiwari, M. Integrated Scheduling of Rake and Stockyard Management with Ship Berthing: A Block Based Evolutionary Algorithm. *Int. J. Prod. Res.* **2016**, *54*, 4182–4204. [\[CrossRef\]](#)
- Pratap, S.; Nayak, A.; Kumar, A.; Cheikhrouhou, N.; Tiwari, M.K. An Integrated Decision Support System for Berth and Ship Unloader Allocation in Bulk Material Handling Port. *Comput. Ind. Eng.* **2017**, *106*, 386–399. [\[CrossRef\]](#)
- Rocha De Paula, M.; Boland, N.; Ernst, A.T.; Mendes, A.; Savelsbergh, M. Throughput Optimisation in a Coal Export System with Multiple Terminals and Shared Resources. *Comput. Ind. Eng.* **2019**, *134*, 37–51. [\[CrossRef\]](#)
- Unsal, O.; Oguz, C. An Exact Algorithm for Integrated Planning of Operations in Dry Bulk Terminals. *Transp. Res. Part Logist. Transp. Rev.* **2019**, *126*, 103–121. [\[CrossRef\]](#)
- Burdett, R.L.; Corry, P.; Yarlagadda, P.K.; Eustace, C.; Smith, S. A Flexible Job Shop Scheduling Approach with Operators for Coal Export Terminals. *Comput. Oper. Res.* **2019**, *104*, 15–36. [\[CrossRef\]](#)
- Burdett, R.L.; Corry, P.; Eustace, C.; Smith, S. A Flexible Job Shop Scheduling Approach with Operators for Coal Export Terminals—A Mature Approach. *Comput. Oper. Res.* **2020**, *115*, 104834. [\[CrossRef\]](#)
- Burdett, R.L.; Corry, P.; Eustace, C. Stockpile Scheduling with Geometry Constraints in Dry Bulk Terminals. *Comput. Oper. Res.* **2021**, *130*, 105224. [\[CrossRef\]](#)
- Gao, Z.; Sun, D.; Zhao, R.; Dong, Y. Ship-Unloading Scheduling Optimization for a Steel Plant. *Inf. Sci.* **2021**, *544*, 214–226. [\[CrossRef\]](#)
- Cao, Z.; Wang, W.; Jiang, Y.; Xu, X.; Xu, Y.; Guo, Z. Joint Berth Allocation and Ship Loader Scheduling under the Rotary Loading Mode in Coal Export Terminals. *Transp. Res. Part Methodol.* **2022**, *162*, 229–260. [\[CrossRef\]](#)
- Zhang, X.; Li, J.; Yang, Z.; Wang, X. Collaborative Optimization for Loading Operation Planning and Vessel Traffic Scheduling in Dry Bulk Ports. *Adv. Eng. Inform.* **2022**, *51*, 101489. [\[CrossRef\]](#)
- Tang, M.; Gong, D.; Liu, S.; Zhang, H. Applying Multi-Phase Particle Swarm Optimization to Solve Bulk Cargo Port Scheduling Problem. *Adv. Prod. Eng. Manag.* **2016**, *11*, 299–310. [\[CrossRef\]](#)
- Tang, L.; Sun, D.; Liu, J. Integrated Storage Space Allocation and Ship Scheduling Problem in Bulk Cargo Terminals. *IIE Trans.* **2016**, *48*, 428–439. [\[CrossRef\]](#)
- Hu, G. Study on Joint Dispatching of Bulk Carriers Berth and Ship Unloader. *Open J. Bus. Manag.* **2018**, *6*, 318–332. [\[CrossRef\]](#)
- Krimi, I.; Benmansour, R.; Cadi, A.A.E.; Deshayes, L.; Duviolier, D.; Elhachemi, N. A Rolling Horizon Approach for the Integrated Multi-Quays Berth Allocation and Crane Assignment Problem for Bulk Ports. *Int. J. Ind. Eng. Comput.* **2019**, *10*, 577–591. [\[CrossRef\]](#)

20. Peng, Y.; Dong, M.; Li, X.; Liu, H.; Wang, W. Cooperative Optimization of Shore Power Allocation and Berth Allocation: A Balance between Cost and Environmental Benefit. *J. Clean. Prod.* **2021**, *279*, 123816. [[CrossRef](#)]
21. Yang, A.; Cao, Y.; Chen, K.; Zeng, Q.; Chen, Z. An Optimization Model for Tramp Ship Scheduling Considering Time Window and Seaport Operation Delay Factors. *J. Adv. Transp.* **2021**, *2021*, 6650097. [[CrossRef](#)]
22. Li, J.; Zhang, X.; Yang, B.; Wang, N. Vessel Traffic Scheduling Optimization for Restricted Channel in Ports. *Comput. Ind. Eng.* **2021**, *152*, 107014. [[CrossRef](#)]
23. Cheimanoff, N.; Fontane, F.; Kitri, M.N.; Tchernev, N. A Reduced VNS Based Approach for the Dynamic Continuous Berth Allocation Problem in Bulk Terminals with Tidal Constraints. *Expert Syst. Appl.* **2021**, *168*, 114215. [[CrossRef](#)]
24. Xu, X.; Wang, W.; Peng, Y.; Song, X. The Optimaztion of Train Collection Strategies in Coal Ports with Eco-Friendly Yards: A Case Study in Nothern China. In Proceedings of the 2017 2nd International Conference Sustainable and Renewable Energy Engineering (ICSREE), Hiroshima, Japan, 10–12 May 2017; pp. 58–61. [[CrossRef](#)]
25. Belov, G.; Boland, N.; Savelsbergh, M.W.P.; Stuckey, P.J. Local Search for a Cargo Assembly Planning Problem. In *Integration of AI and OR Techniques in Constraint Programming*; Springer International Publishing: Cham, Switzerland, 2014; Volume 8451, pp. 159–175. [[CrossRef](#)]
26. Savelsbergh, M.; Smith, O. Cargo Assembly Planning. *EURO J. Transp. Logist.* **2015**, *4*, 321–354. [[CrossRef](#)]
27. Mao, Y.; Zhang, L. Design and Implementation of Port Bulk Storage Management System Based on Internet of Things Technology. *J. Coast. Res.* **2019**, *98*, 62–66. [[CrossRef](#)]
28. Angelelli, E.; Kalinowski, T.; Kapoor, R.; Savelsbergh, M.W.P. A Reclaimer Scheduling Problem Arising in Coal Stockyard Management. *J. Sched.* **2016**, *19*, 563–582. [[CrossRef](#)]
29. Kalinowski, T.; Kapoor, R.; Savelsbergh, M.W.P. Scheduling Reclaimers Serving a Stock Pad at a Coal Terminal. *J. Sched.* **2017**, *20*, 85–101. [[CrossRef](#)]
30. Guo, Z.; Cao, Z.; Wang, W.; Jiang, Y.; Xu, X.; Feng, P. An Integrated Model for Vessel Traffic and Debballasting Scheduling in Coal Export Terminals. *Transp. Res. Part Logist. Transp. Rev.* **2021**, *152*, 102409. [[CrossRef](#)]
31. Belov, G.; Boland, N.L.; Savelsbergh, M.W.P.; Stuckey, P.J. Logistics Optimization for a Coal Supply Chain. *J. Heuristics* **2020**, *26*, 269–300. [[CrossRef](#)]
32. Tuson, A.; van Hentenryck, P. The OPL Optimization Programming Language. *J. Oper. Res. Soc.* **2000**, *51*, 649–650. [[254202](#)]. [[CrossRef](#)]
33. Laborie, P.; Rogerie, J.; Shaw, P.; Vilím, P. IBM ILOG CP Optimizer for Scheduling: 20+ Years of Scheduling with Constraints at IBM/ILOG. *Constraints* **2018**, *23*, 210–250. [[CrossRef](#)]
34. Lazaar, N.; Gotlieb, A.; Lebbah, Y. A CP Framework for Testing CP. *Constraints* **2012**, *17*, 123–147. [[CrossRef](#)]
35. Zheng, Q.Q.; Zhang, Y.; He, L.J.; Tian, H.W. Discrete Multi-Objective Artificial Bee Colony Algorithm for Green Co-Scheduling Problem of Ship Lift and Ship Lock. *Adv. Eng. Inform.* **2023**, *55*, 101897. [[CrossRef](#)]
36. 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. [[CrossRef](#)]
37. Khichane, M.; Albert, P.; Solnon, C. Strong Combination of Ant Colony Optimization with Constraint Programming Optimization. In Proceedings of the Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Bologna, Italy, 14–18 June 2010 ; Lodi, A., Milano, M., Toth, P., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; pp. 232–245. [[CrossRef](#)]
38. Mohri, S.S.; Asgari, N.; Zanjirani Farahani, R.; Bourlakis, M.; Laker, B. Fairness in Hazmat Routing-Scheduling: A Bi-Objective Stackelberg Game. *Transp. Res. Part Logist. Transp. Rev.* **2020**, *140*, 102006. [[CrossRef](#)]
39. Iris, Ç.; Pacino, D.; Ropke, S. Improved Formulations and an Adaptive Large Neighborhood Search Heuristic for the Integrated Berth Allocation and Quay Crane Assignment Problem. *Transp. Res. Part Logist. Transp. Rev.* **2017**, *105*, 123–147. [[CrossRef](#)]
40. Iris, Ç.; Pacino, D.; Ropke, S.; Larsen, A. Integrated Berth Allocation and Quay Crane Assignment Problem: Set Partitioning Models and Computational Results. *Transp. Res. Part Logist. Transp. Rev.* **2015**, *81*, 75–97. [[CrossRef](#)]
41. Hottenrott, A.; Waidner, L.; Grunow, M. Robust Car Sequencing for Automotive Assembly. *Eur. J. Oper. Res.* **2021**, *291*, 983–994. [[CrossRef](#)]
42. Ropke, S.; Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transp. Sci.* **2006**, *40*, 455–472. [[CrossRef](#)]
43. Yilmazlar, I.O.; Kurz, M.E. Adaptive Local Search Algorithm for Solving Car Sequencing Problem. *J. Manuf. Syst.* **2023**, *68*, 635–643. [[CrossRef](#)]
44. Caponio, A.; Cascella, G.L.; Neri, F.; Salvatore, N.; Sumner, M. A Fast Adaptive Memetic Algorithm for Online and Offline Control Design of PMSM Drives. *IEEE Trans. Syst. Man, Cybern. Part (Cybern.)* **2007**, *37*, 28–41. [[CrossRef](#)]
45. Ong, Y.S.; Lim, M.H.; Chen, X. Memetic Computation—Past, Present & Future [Research Frontier]. *IEEE Comput. Intell. Mag.* **2010**, *5*, 24–31. [[CrossRef](#)]
46. Brandão, J. Iterated Local Search Algorithm with Ejection Chains for the Open Vehicle Routing Problem with Time Windows. *Comput. Ind. Eng.* **2018**, *120*, 146–159. [[CrossRef](#)]
47. Zhou, Y.; Wang, J. A Local Search-Based Multiobjective Optimization Algorithm for Multiobjective Vehicle Routing Problem With Time Windows. *IEEE Syst. J.* **2015**, *9*, 1100–1113. [[CrossRef](#)]

48. Ribas, I.; Companys, R.; Tort-Martorell, X. Efficient Heuristics for the Parallel Blocking Flow Shop Scheduling Problem. *Expert Syst. Appl.* **2017**, *74*, 41–54. [[CrossRef](#)]
49. Salama, M.; Srinivas, S. Adaptive Neighborhood Simulated Annealing for Sustainability-Oriented Single Machine Scheduling with Deterioration Effect. *Appl. Soft Comput.* **2021**, *110*, 107632. [[CrossRef](#)]
50. Fathollahi-Fard, A.M.; Govindan, K.; Hajiaghayi-Keshteli, M.; Ahmadi, A. A Green Home Health Care Supply Chain: New Modified Simulated Annealing Algorithms. *J. Clean. Prod.* **2019**, *240*, 118200. [[CrossRef](#)]
51. Puchta, M.; Gottlieb, J. Solving Car Sequencing Problems by Local Optimization. In Proceedings of the Applications of Evolutionary Computing, Kinsale, Ireland, 3–4 April 2002; Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2002; pp. 132–142. [[CrossRef](#)]
52. Hansen, P.; Mladenović, N.; Urošević, D. Variable Neighborhood Search and Local Branching. *Comput. Oper. Res.* **2006**, *33*, 3034–3045. [[CrossRef](#)]
53. Steinbrunn, M.; Moerkotte, G.; Kemper, A. Heuristic and Randomized Optimization for the Join Ordering Problem. *VLDB J.* **1997**, *6*, 191–208. [[CrossRef](#)]
54. Hao, L.; Jin, J.G.; Zhao, K. Joint Scheduling of Barges and Tugboats for River–Sea Intermodal Transport. *Transp. Res. Part Logist. Transp. Rev.* **2023**, *173*, 103097. [[CrossRef](#)]
55. Xu, Y.; Qi, L.; Luan, W.; Guo, X.; Ma, H. Load-In-Load-Out AGV Route Planning in Automatic Container Terminal. *IEEE Access* **2020**, *8*, 157081–157088. [[CrossRef](#)]
56. Pellerin, R.; Perrier, N.; Berthaut, F. A Survey of Hybrid Metaheuristics for the Resource-Constrained Project Scheduling Problem. *Eur. J. Oper. Res.* **2020**, *280*, 395–416. [[CrossRef](#)]
57. Iris, Ç.; Lam, J.S.L. A Review of Energy Efficiency in Ports: Operational Strategies, Technologies and Energy Management Systems. *Renew. Sustain. Energy Rev.* **2019**, *112*, 170–182. [[CrossRef](#)]

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