Hierarchical Planning and Scheduling for Bulk Ports via Network Flow and Deep Reinforcement Learning-Guided Constraint Programming

Xuan Lu[a,b], *Yu Zhang*[a], *Xuri Xin*[b], *Hang Yang*[c], *Huanhuan Li*[b,d], *Lanbo Zheng*[a,*], *Zaili Yang*[b,*]

[a]*School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, PR China*
[b]*Liverpool Logistics, Offshore and Marine Research Institute, Liverpool John Moores University, Liverpool L3 3AF, UK*
[c]*School of Information Engineering, Wuhan University of Technology, Wuhan, 430070, PR China*
[d]*School of Engineering, University of Southampton, Southampton, SO17 1BJ, UK*

ARTICLEINFO

Keywords:
Dry bulk cargo port
Port management
Yard equipment scheduling
Deep reinforcement learning
Constraint programming

ABSTRACT

In this research, an integrated inbound and outbound operational planning and scheduling problem is addressed for complex and large bulk ports. The practice of moving homogeneous dry bulk cargoes on a fixed terminal is changing as raw materials of different types are transported from/to the same terminals. It raises a new research challenge where unloading, stacking, reclaiming, conveying and loading operations must be coordinated to import/export blended products according to the tight specifications of customers. This paper aims to maximise resource utilisation and to satisfy demands as early as possible. The essence of the problem is to design the routing of product flows throughout the port logistics network such that supply and demand are matched optimally. This study presents a new framework that enables the modelling of the planning part as a multi-commodity flow problem and the scheduling part as a constraint programming (CP) problem. A novel dual-engine optimisation method that synergistically combines CP with deep reinforcement learning (DRL) is proposed to accelerate the scheduling phase. The method leverages DRL agents to fix key variables, thereby effectively accelerating the optimisation process of the CP solver. Comprehensive numerical experiments are conducted on real data sets as well as instances derived from real scenarios to validate the effectiveness of the proposed approach, demonstrating significant improvements in port scheduling efficiency. Additionally, strategic management analyses offer actionable insights to support decision-making in bulk port operations. The proposed methods provide a generalised methodology adaptable to a broad range of complex combinatorial optimisation problems in port logistics and beyond, paving the way for more intelligent and sustainable dry bulk port management.

1. Introduction

Shipping is a critical component of the global logistics and transportation system, serving as a vital facilitator for international trade and economic connectivity (Li et al., 2024, 2023). Within this system, ports play a central role, linking inland transportation networks to ocean trade routes and supporting economic growth through enhanced trade efficiency and accessibility (Cai et al., 2024b, 2024a). Among them, dry bulk ports specialise in handling essential bulk commodities such as coal, ore, and grain, which are vital for energy security and industrial production (Ernst et al., 2017). In response to increasing global demand for bulk cargo, major international dry bulk ports have undergone infrastructure expansion, evolving from single-functional facilities handling a single cargo type (Le Carrer et al., 2020; Cao et al., 2022; Gao et al., 2024) to large-scale, multi-cargo, multi-functional ports (Jiang et al., 2024). This transformation aims to improve loading and unloading capacities, operational efficiency, and environmental performance (Fang et al., 2021). The presence of multiple cargo types (e.g., coal and ore) and diverse operational functions (e.g., train and ship loading/unloading) introduces significant complexity to port management. Multi-cargo operations increase equipment variety and quantity, with operational capabilities differing across cargo types. Multi-functional operations result in diverse cargo flows and more complex resource conflicts. Cargo mixing operations in cargo-assembly dry bulk ports (CDBPs), blending raw materials to meet customer specifications, introduce complex constraints that challenge coordinated scheduling. These factors jointly complicate the global scheduling of bulk ports, which must integrate train scheduling, yard management, equipment allocation, and ship scheduling. With the rise of new scenarios and increasing complexity, traditional manual and experience-based scheduling methods have evidenced

inadequacies, making it difficult to generate effective solutions in a reasonable timeframe. As a result, manual scheduling struggles to meet the evolving demands of modern bulk ports, ultimately becoming a bottleneck for enhancing their efficiency and service levels.

The increasing adoption of automated handling equipment and digitalised port management has enabled higher levels of electrification and automation in dry bulk ports (Cao et al., 2025). However, compared to container ports, dry bulk ports face more significant challenges due to continuous cargo mixing operations, diverse cargo types, and complex loading and unloading processes. Despite these challenges, they still lag in the adoption of intelligent scheduling systems and digital management technologies (Zhang et al., 2024), hindering their ability to transition toward fully automated, efficient, and sustainable operations. Previous studies (Lu et al., 2024; Pratap et al., 2018) on scheduling continuous operation lines in dry bulk ports have primarily concentrated on single-cargo scheduling, overlooking the broader challenges of integrated production planning and operation scheduling in multi-cargo, multi-functional dry bulk ports. To bridge this gap, new models and algorithms need to be developed to optimise production efficiency and service quality in increasingly complex bulk port environments.

This paper addresses two primary challenges in CDBP operations.

Challenge 1: How can an integrated solution for production planning and equipment scheduling be developed to efficiently manage diverse cargo types and multiple loading and unloading directions?

Challenge 2: How can an effective and scalable optimisation framework be designed to solve large-scale planning and scheduling problems within practical computation times?

To address **Challenge 1**, this paper proposes a hierarchical solution framework that decomposes the complex production planning and scheduling problem into two interconnected sub-problems: production planning and operational task scheduling. This structured decomposition transforms the port's operational challenges into a combinatorial optimisation problem that integrates network flow modelling and equipment scheduling constraints. However, directly solving this problem with mixed-integer programming (MIP) is impractical as it involves decisions at different levels of time granularity. Instead, the proposed hierarchical approach ensures problem consistency while leveraging specialised optimisation techniques to balance model realism, solution accuracy, and computational efficiency.

To tackle **Challenge 2**, this study introduces a novel dual-engine optimisation approach that accelerates the solution of complex scheduling problems by integrating deep reinforcement learning (DRL) with constraint programming (CP). A deep Q-network (DQN) is employed to fix key decision variables, guiding the CP solver to accelerate convergence and enhance optimisation performance. By leveraging prior knowledge from training data, the DQN agent extracts historical strategies to support the CP solver, effectively balancing solution quality and computational efficiency for real-time port scheduling.

In summary, this paper makes three primary contributions to the field of bulk port production planning and scheduling:

(1) Development of a hierarchical optimisation framework for integrated multi-cargo, multi-functional bulk port planning and scheduling, effectively decomposing complex port operations into manageable sub-problems.

(2) Introduction of a dual-engine optimisation approach, combining DRL and CP to achieve high-quality, efficient scheduling solutions within practical computation times.

(3) Demonstration of significant efficiency gains in bulk port operations, validated through comprehensive numerical experiments, offering insights into strategic port management and intelligent decision-making.

The structure of this paper is organised as follows: Section 2 systematically reviews the literature, outlining the current research status and primary methods in dry bulk port optimisation, identifying research gaps, and presenting the contributions. Section 3 details the research problem. Section 4 introduces the hierarchical solution framework and its mathematical model.

Section 5 describes the general solution approach and the dual-engine-driven optimisation method. Section 6 presents and discusses the experimental results, along with the impact of different port management strategies. Finally, Section 7 concludes the research findings, discusses limitations and provides insights into future research.

2. Literature review

This section systematically reviews current research on bulk port planning and scheduling, identifies key methodological approaches, and highlights existing gaps.

2.1. A systematic literature review

In this section, a general literature review is carried out to obtain a comprehensive overview of the integrated planning and scheduling problem of bulk cargo ports. Using the Web of Science (WoS) core database to retrieve peer-reviewed journal papers from 1985 to 2024, 491 papers were obtained. After carefully reviewing the titles, keywords, abstracts, and section titles of the papers, 66 articles focusing on bulk port planning and scheduling problems were selected for further analysis. An analysis of published papers and their research topics over the years was conducted. Utilising the CiteSpace tool, keyword cluster analysis and keyword development trend analysis were performed on the refined papers. The specific steps involved are illustrated in Fig. 1.

Analysis of the number of published papers reveals a consistent upward trend in research on bulk port planning and scheduling problems, reflecting sustained scholarly interest. The complexity of bulk port operations results in a broad range of research topics, with numerous studies addressing each subcomponent of the bulk port system. The primary focus areas include berth allocation and shore equipment scheduling problems (Pratap et al., 2017; Krimi et al., 2020; Cao et al., 2022), which are closely interconnected. Additionally, the joint scheduling of stockyard management and loading/unloading operations has garnered attention (Cimpeanu et al., 2017; Pratap et al., 2018, 2016), although these issues present significant challenges due to the intricacies of the bulk port production process and the diversity of logistics entities involved. Vessel scheduling and berth allocation problems also continue to attract considerable scholarly interest (Guo et al., 2021).

Keyword clustering results indicate that scholarly research predominantly centres on ‘loading operation planning’ and ‘berth allocation’. Given the complexities associated with bulk port planning and scheduling problems, many researchers advocate for hierarchical methods to decompose these intricate problems. Solutions typically include simulation, approximation algorithms, exact algorithms, and intelligent optimisation algorithms. An analysis of research trends from 2014 to 2024 reveals that studies primarily concentrate on model formulation and algorithm development for specific production components, while the challenges of global production and scheduling optimisation in dry bulk ports remain largely underexplored. Furthermore, advancements in machine learning technologies, particularly DRL, offer novel solutions for managing dry bulk ports. Therefore, leveraging new technologies to develop effective global planning and scheduling strategies for dry bulk ports is both reasonable and innovative.

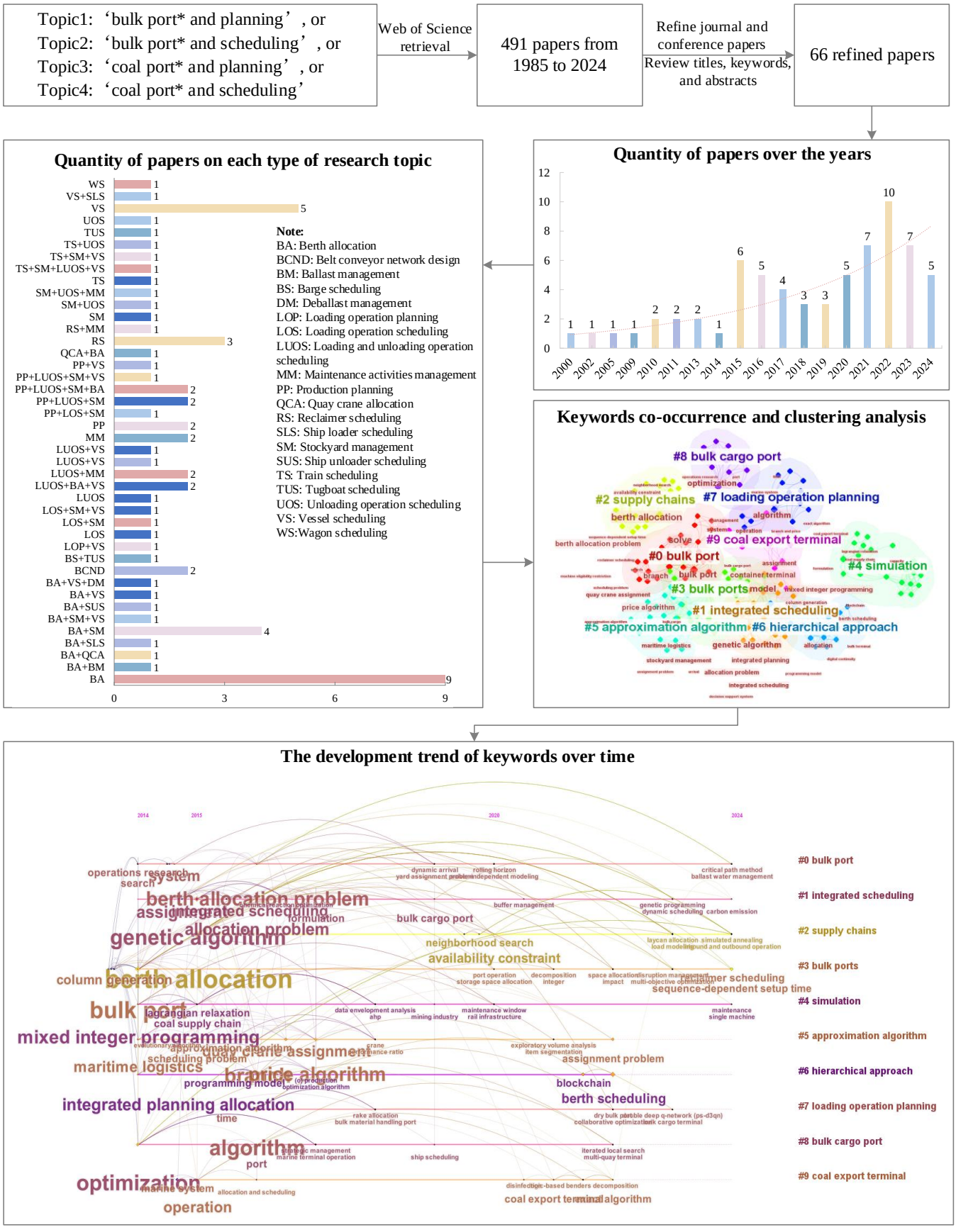


Fig. 1. The systematic literature review procedure and results.

2.2. Bulk port planning and scheduling problem

This section outlines the research focus and challenges of bulk port planning and scheduling from the perspectives of single-resource and multi-resource integrated optimisation, highlighting the novelty of this study through a review of nine closely related articles.

Single-resource scheduling optimisation: The single-resource scheduling problem at bulk ports primarily encompasses equipment scheduling, resource allocation, and path planning. In the realm of train scheduling, Xu et al. (2017) utilised a simulation model to enhance the train assembly strategy at coal ports, demonstrating that shorter train assembly cycles significantly reduce operational costs. For the belt routing problem, Mallah et al. (2023) established a mixed-integer linear programming (MILP) model to address the dry bulk port belt routing challenge. For stockyard management, Belov et al. (2014, 2015) and Savelsbergh and Smith (2015) developed hybrid algorithms based on large neighbourhood search and tree search to optimise coal stockpile assembly processes, thereby improving overall operational efficiency at coal ports. Regarding stockyard equipment scheduling, Angelelli et al. (2016) established stacker-reclaimer operation models under varying scenarios, analysed the problem's complexity under different constraints, and proposed a pseudo-polynomial algorithm for scheduling two stacker-reclaimers with specified storage locations and picking orders. For the berth scheduling problem, Cheimanoff et al. (2021) introduced a reduced variable neighbourhood search method to address the dynamic continuous berth allocation problem specific to coal ports. Additionally, Li et al. (2021) formulated a multi-objective optimisation model that incorporates various traffic constraints for managing vessel traffic in shared channels within coal ports.

Multi-resource integrated scheduling optimisation: Research on multi-resource integrated scheduling optimisation in bulk ports primarily emphasises the integrated optimisation of multiple operational processes or scenarios. In the realm of integrated scheduling problems of stockyard and seaside equipment, L. Tang et al. (2016) addressed the optimisation challenge of ship unloading operations and stockyard allocation at bulk cargo terminals in large steel enterprises by formulating an MIP model. For multi-resource integrated optimisation at the seaside, Krimi et al. (2019) introduced an efficient heuristic algorithm based on the rolling horizon method to tackle the integrated problem of berth allocation and crane assignment in bulk cargo ports. Additionally, Cao et al. (2022) developed an MIP model for the berth allocation and ship loader scheduling problem at a coal port, successfully deriving an exact solution through logic-based Benders decomposition techniques. In the realm of integrated optimisation between the seaside and vessel traffic, Wang et al. (2023) introduced an interference management strategy alongside a two-stage row generation algorithm to address the collaborative scheduling of ships and ship loaders. Regarding port collection, distribution, and stockyard integrated optimisation, Babu et al. (2015) developed two heuristic-based greedy construction algorithms aimed at addressing stockyard management, train scheduling, and vessel scheduling challenges in export coal ports. Additionally, Menezes et al. (2016, 2017) formulated an integrated model that considered both stockyard management and loading and unloading operations for operational planning and scheduling of ore ports. Burdett et al. (2019, 2020, 2021) conducted extensive research on the production and operational management of export coal ports, covering critical areas such as berth scheduling, ship loader scheduling, stacker-reclaimer scheduling, and yard inventory management. Belov et al. (2020) developed an integrated model encompassing train scheduling, stockyard management, and vessel scheduling, comparing the performance of the MIP model with that of the CP model.

After a comprehensive review of the bulk port planning and scheduling problem, this section selects nine papers focusing on the integrated scheduling of bulk ports, chosen based on citations and journal quality. These papers are analysed in depth regarding their research objectives, content, and methodologies. Table 1 provides a comparative overview of the research elements in these nine papers alongside this study.

Table 1

Literature review on integrated planning and scheduling of bulk ports.

Refs	Number of operational directions considered	Cargo type	Production planning	Landside management			Stockyard management		Seaside management		Maritime management		Problem abstraction	Methods
				TS	DS	LS	SM	SRS	BA	SES	VTS	VD		
Umang et al. (2013)	2	Coal, grain and oil							✓	✓			GSPP	MIP+SWO
Menezes et al. (2017)	2	Ore	✓		✓		✓	✓		✓			PFPSP	CG+B&P
Burdett et al. (2020)	2	Coal					✓	✓		✓		✓	FJSP	SA
Belov et al. (2020)	2	Coal	✓	✓			✓	✓	✓	✓	✓		—	CP+LNS+RBH
Bouzekri et al. (2022)	1	Fertiliser	✓				✓	✓	✓	✓	✓		—	MIP
Zhang et al. (2022)	1	Coal						✓		✓	✓		—	VNS+ NSGA-II
Wang et al. (2023)	1	Coal							✓	✓	✓		—	RG
de Andrade and Menezes (2023)	2	Coal	✓				✓	✓	✓	✓			—	CG+DH
Gao et al. (2024)	2	Ore					✓	✓	✓	✓			—	CG
This study	4	Coal, ore and aluminium	✓	✓	✓	✓	✓	✓		✓	✓		MCF+FJSP	CP+DQN

Notes: TS: Train scheduling; DS: Dump station scheduling; LS: Loading station scheduling; SM: Stockyard management; SRS: Stacker/reclaimer scheduling; BA: Berth allocation; SES: Seaside equipment scheduling; VTS: Vessel traffic scheduling; VD: Vessel deballast; GSPP: Generalised set partitioning problem; PFPSP: Product flow planning and scheduling problem; FJSP: Flow job shop problem; MCF: Multi-commodity flow problem; MIP: Mixed-integer programming; SWO: Squeaky wheel optimisation; CG: Column generation; B&P: Branch and price algorithm; SA: Simulated annealing algorithm; CP: Constraint programming; LNS: Large neighbourhood search; RBH: Rule-based heuristic; VNS: Variable neighbourhood search; NSGA-II: Non-dominated sorting genetic algorithm II; RG: Row generation; DH: Diving heuristic; DQN: Deep Q-network; ✓: Yes; —: Not mentioned.

Based on Table 1 and the above review, the following conclusions can be drawn:

(1) While notable progress has been made in bulk port planning and scheduling research, most existing studies on integrated planning and scheduling focus on single-cargo scenarios with limited operational complexity. In contrast, the scenario of integrated planning and scheduling for multi-cargo, multi-functional dry bulk ports, which is considerably more complex and represents a frequently encountered real-world issue, remains largely underexplored. Although some scholars have studied multiple cargo types (Umang et al., 2013; Robenek et al., 2014), their focus remains on vessel service time, without addressing port throughput or discussing coal blending operations in bulk ports.

(2) While some scholars have recognised the continuity between port production planning and scheduling (Menezes et al., 2017; de Andrade and Menezes, 2023), few have incorporated continuous production scheduling into the integrated bulk port problem. Additionally, limited research addresses the integrated scheduling of trains, vessels, and various loading and unloading equipment. This study offers a more holistic approach by integrating port production planning and scheduling to optimise the scheduling of trains, vessels, and loading and unloading processes. It is notable that, based on port surveys, vessel traffic and deballasting scheduling are managed independently by different production teams, and are therefore excluded from this study.

(3) Due to the complexity of bulk port operations, most studies directly model these workflows (Menezes et al., 2016), typically assuming that port operation plans have already been established. This overlooks the interdependence between the planning and scheduling stages. To address this limitation, this study introduces a new operational scenario based on real cases from large-scale dry bulk ports, integrating production planning and operation scheduling into a unified framework. By decomposing the problem into several classical optimisation sub-problems, a more structured and modular analysis framework is developed. This decomposition also enhances the generalisability of the proposed model and solution approach to other engineering domains with similar decision-making complexities.

(4) Most scholars have used exact or heuristic algorithms to solve bulk port planning and scheduling problems. However, challenges persist in addressing large-scale problems and certifying the feasibility or optimality of solutions. This highlights the need to develop new efficient solving algorithms to enhance solution quality. This paper proposes a dual-engine-driven solution method that combines CP with DQN, striking a balance between solution efficiency and quality. The method's feasibility and generality are validated through three real-world port operation cases and multi-scale instance experiments. This new solution framework offers a novel approach to addressing dry bulk port management challenges.

2.3. Methods employed

The mentioned literature reveals that current research on the production planning and scheduling of bulk ports primarily abstracts port production scenarios and operations into mathematical or simulation models, followed by the development of corresponding solution algorithms or heuristic strategies. The main approaches include exact algorithms, approximation algorithms, intelligent optimisation algorithms, and simulation techniques. Furthermore, with the advancement of deep learning and reinforcement learning technologies, some scholars have begun incorporating these methods into the management of bulk ports, expanding the scope of potential optimisation strategies.

Mathematical models and exact algorithms: Given that bulk ports are complex systems characterised by numerous logistical objects and intricate operational constraints, scholars have increasingly employed integrated models to describe equipment coordination and loading/unloading processes. Gao et al. (2021) framed the loading operations at a steel plant's bulk terminal as a flexible job shop scheduling problem, formulating a corresponding MIP model and a column generation algorithm. Bouzekri et al. (2023) established an integer programming (IP) model for the integrated problem of berth allocation and vessel scheduling, factoring in equipment constraints and preventive maintenance, and employed the Xpress solver for model resolution. de Andrade and Menezes (2023) addressed the integrated scheduling of operational planning, yard allocation, and berth allocation in dry bulk ports, formulating an MILP model resolved through a column generation framework that integrates backtracking algorithms, relaxed fixed methods, and the CPLEX solver. Gao et al. (2024) proposed a two-step model framework for the berth allocation and mining operation problem, utilising column generation and infeasibility cut correction.

Heuristic algorithms: Considering the complexity of bulk port production scenarios and the limited capacity of exact algorithms to address large-scale problems, some researchers have explored the use of heuristic algorithms for more efficient solutions. de Andrade and Menezes (2021) addressed the collaborative allocation of unloading operations and berthing in dry bulk ports by establishing an MILP model and designing a hybrid algorithm incorporating rolling horizon, backtracking heuristics, and relaxed fixed heuristics. Guo et al. (2021) proposed a comprehensive scheduling model for vessel traffic and ballast management in bulk ports, demonstrating its efficiency compared to heuristic rules like first-in-first-out (FIFO). Zhang et al. (2022) developed a multi-objective mathematical model for the integrated optimisation of loading operations and vessel traffic at dry bulk ports, solving the model using a combination of variable neighbourhood search (VNS) and non-dominated

sorting genetic algorithm II (NSGA-II). Gao et al. (2022) designed a differential evolution algorithm for unloading operations at large steel mills. Cheimanoff et al. (2022) proposed an MILP model for berth allocation at bulk terminals and implemented a combination of greedy algorithms and iterated local search (ILS) for practical problem resolution. Ferreira et al. (2023) tackled deterministic and uncertain scheduling issues in unloading operations, utilising MIP models, CP models, greedy stochastic algorithms, and genetic programming for resolution.

Simulation: To validate the effectiveness and feasibility of optimisation methods for dry bulk port production organisations, researchers often employ simulation technologies to replicate and assess the production processes and management strategies at these ports. Cimpeanu et al. (2017) and Neagoe et al. (2021) respectively utilised discrete-event simulation to simulate the unloading processes of bulk cargo ports and to analyse the effects of various management measures on critical performance indicators, such as port throughput and truck waiting time. Mohajeri et al. (2021) proposed a hybrid simulation approach combining multi-body dynamics and the discrete element method to model bulk cargo handling equipment, enabling performance enhancements and operational optimisation.

Machine learning: Due to the complexity of bulk cargo port management and the lack of sufficient training data, directly solving the bulk port planning and scheduling problems using machine learning methods poses a significant challenge. Therefore, most scholars apply machine learning techniques for prediction or algorithm parameter tuning. Štepec et al. (2020) and Chu et al. (2024) introduced machine learning-based approaches to predict the turnover time of vessels at bulk ports, leveraging port docking data for improved accuracy. Li et al. (2022) created an adaptive genetic simulated annealing algorithm tailored for ship scheduling challenges, framing the parameter selection process as a Markov decision process (MDP) and employing reinforcement learning to dynamically select algorithm parameters. Furthermore, C. Li et al. (2022) proposed a scheduling state selection method grounded in DRL to enhance the scheduling efficiency at bulk cargo terminals. To accurately predict the arrival times of bulk carriers, EI Mekkaoui et al. (2023) introduced a method utilising a deep learning sequence model, effectively leveraging historical navigation data to improve arrival time forecasts. Ai et al. (2023) abstracted the unloading process of bulk ports as an MDP and utilised a prioritised experience replay and softmax strategy-based duelling double deep Q-network (PS-D3QN) to optimise berth allocation and yard scheduling in bulk cargo ports.

In summary, recent advancements in the optimisation of dry bulk port production organisations have significantly enhanced port operation efficiency and service quality. Early research predominantly concentrated on constructing mathematical models and developing algorithmic solutions. However, in recent years, scholars have started to incorporate machine learning into dry bulk port production, albeit with applications primarily limited to parameter optimisation and prediction. Nevertheless, traditional approaches to mathematical model construction and algorithm solving often focus narrowly on problem structures, failing to leverage prior data and empirical rules effectively. This limitation hampers the ability to optimise production and operational issues in large-scale or uncertain contexts. On the other hand, some scholars have achieved notable solution efficiency by combining operations research optimisation techniques with machine learning methods to address traditional optimisation problems, such as the travelling salesman problem (Cappart et al., 2021), job-shop scheduling problem (Tassel et al., 2023) and 3-D bin packing problem (Jiang et al., 2023). This approach offers new insights into combinatorial optimisation challenges. However, applying DRL directly to large-scale problems often results in exponential network complexity, hindering timely and efficient decision-making (X. Wang et al., 2023). Some studies address train scheduling by decomposing it into sub-problems and applying dynamic optimisation and state update rules (Li and Ni, 2022; X. Wang et al., 2023). However, their applications to large-scale global optimisation remain limited, with insufficient attention to complex constraint handling.

Thus, integrating traditional optimisation with machine learning, leveraging historical data and domain knowledge, and developing a model and data-driven optimisation framework for dry bulk port operations remains a valuable yet challenging research direction.

3. Problem description and formulation

A typical CDBP is depicted in Fig. 2, comprising stockyards, trains, vessels, belt lines, stackers, reclaimers, ship loaders, ship unloaders, loading stations, dump stations and gantry cranes. Stockpiles facilitate the storage of dry bulk goods such as coal and ore. Stackers and reclaimers are responsible for stacking (increasing stockpile inventory) and reclaiming (reducing stockpile inventory) operations. Dump stations and loading stations handle unloading and loading operations for trains, gantry cranes and ship unloaders manage unloading operations for vessels, while ship loaders are responsible for loading operations of vessels. Belt lines connect different operational equipment and areas, serving as the site for cargo mixing operations. The key challenge in planning and scheduling large-scale CDBPs lies in optimising operational volume and efficiency while accounting for the port's actual production layout and processes.

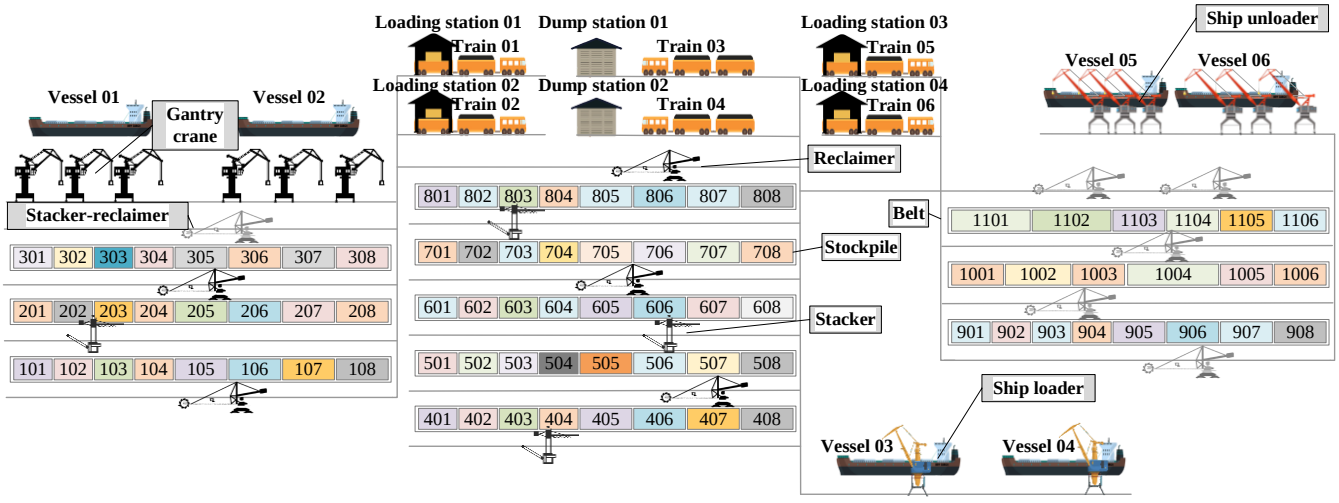


Fig. 2. Schematic layout and equipment connection of a multi-cargo, multi-functional dry bulk port.

The integrated planning and scheduling problem (IPSP) of a large-scale CDBP consists of two main components: operation plan generation and operation task scheduling. In practice, port planners develop an operation plan (as shown in Table 2) by decomposing the loading and unloading demands of arriving ships and trains into multiple operation tasks, based on yard layout, inventory levels, equipment availability, and customer service requirements. Then, the port dispatcher allocates appropriate equipment based on the port layout and resource capacity, aiming to complete all operation tasks efficiently while satisfying operational constraints. Since the completion of each operation task requires coordination between loading/unloading and handling equipment, this study defines an operation stream as an effective combination of equipment capable of collaboratively completing a task. As illustrated in Fig. 3. (a), a single task may correspond to multiple feasible operation streams. Operational constraints refer to requirements in bulk port operations, such as maintaining loading and unloading sequences to prevent capsizing or blending cargoes in fixed ratios to meet customer specifications, which is why blending tasks are included in Table 2. A detailed explanation of these constraints is provided below. Accordingly, the IPSP can be defined as follows: Within a given planning period, assuming all resources operate normally, generate an operation plan for all inbound and outbound vessels and trains based on equipment capacity and accessibility. Considering operational constraints, suitable operation streams are then selected and scheduled to fulfil all demands promptly and minimise operation time.

Table 2

An example of the generated operation plan.

Operation Task	Blending operation task	Task type	Stockpile	Operation volume (ton)
T1-1	/	Inbound	Stockpile 504	4320
T2-1	T2-2	Outbound	Stockpile 202	1500
T2-2	T2-1	Outbound	Stockpile 207	3000
V1-1-1-1	/	Inbound	Stockpile 1101	4100
V2-1-1-1	V2-1-1-2	Outbound	Stockpile 405	1300
V2-1-1-2	V2-1-1-1	Outbound	Stockpile 408	2600

Notes: The first column represents the task name. For instance, "T1-1" denotes task 1 of train 01, while "V1-1-1-1" indicates task 1 of the first operation batch of cabin 1 of vessel 01. The second column specifies the associated task requiring blending with the current operation task. The third column delineates the operation type. The fourth column indicates the designated reclaiming/stacking stockpile. The fifth column represents the workload for the task.

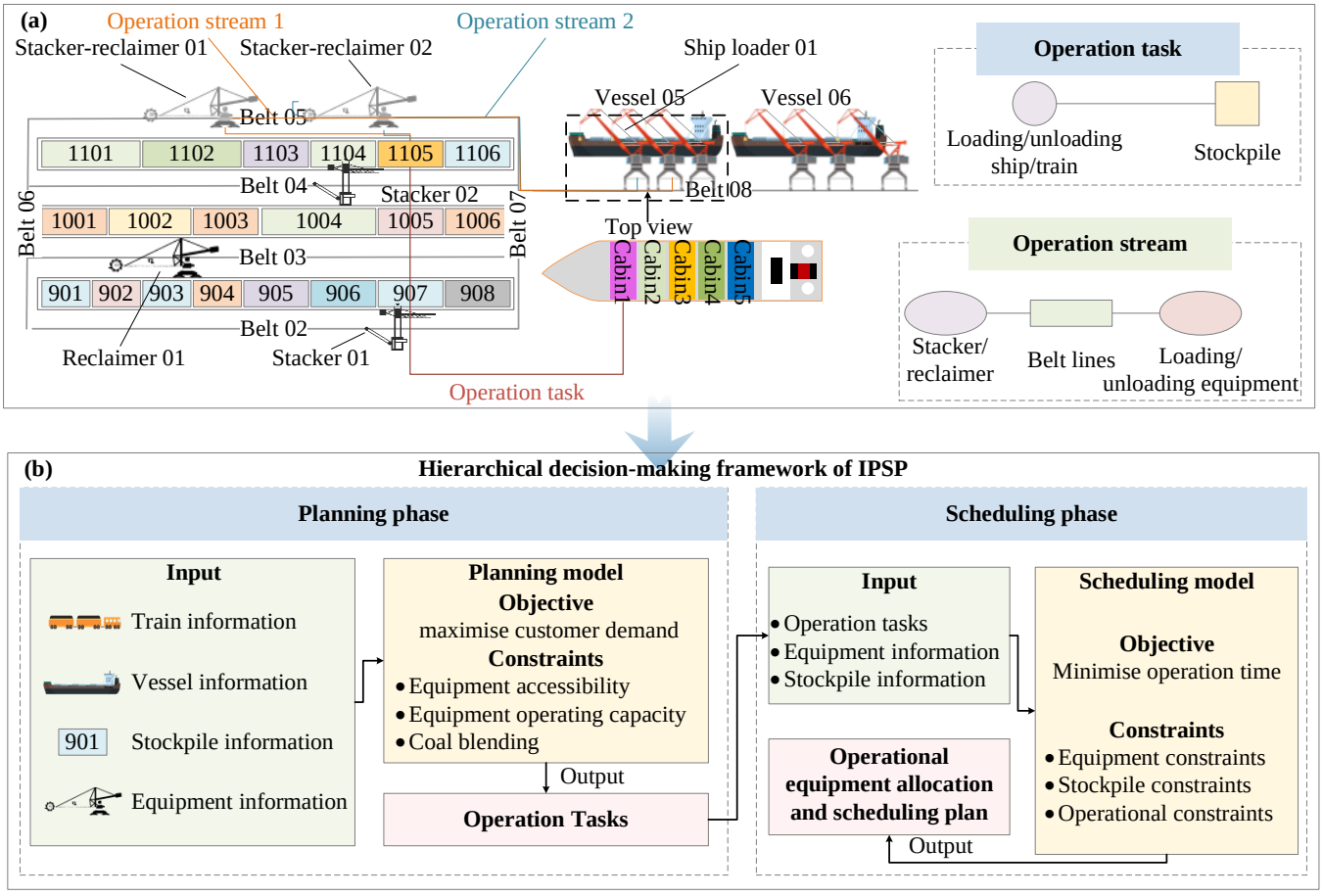


Fig. 3. Illustration of the operation process of IPSP in a multi-cargo, multi-functional dry bulk port.

As a practical problem derived from a multi-cargo, multi-functional dry bulk port, the IPSP entails adhering to various and intricate operational constraints. As shown in Fig. 4, it involves complex elements such as multi-commodity flow planning, resource allocation, flow shop scheduling, and vehicle routing problems. In the forthcoming segments, this article will elucidate the composite nature of the IPSP by integrating a variety of constraints.

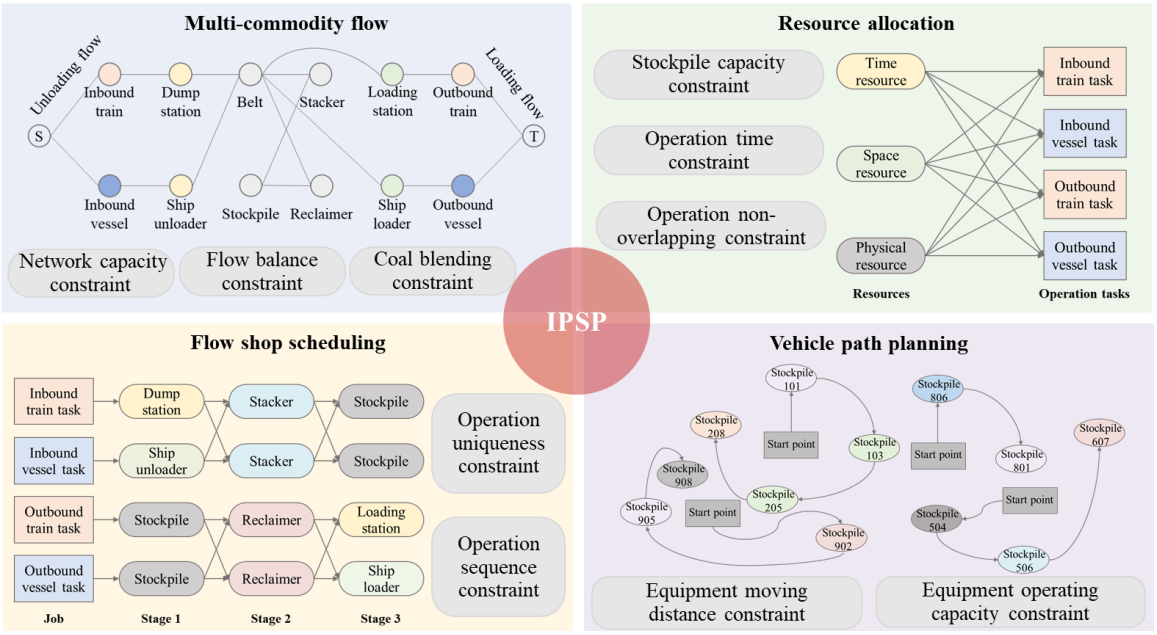


Fig. 4. Illustration of the properties of combinatorial optimisation problems for IPSP.

The MCF problem (Derks and Tijs, 1985) is a classical optimisation problem that revolves around network structures. Its objective is to achieve optimal flow distribution of different commodities flowing through the distribution network by ensuring flow conservation at each node and imposing additional constraints on the flow along the network's arcs. Notably, the planning phase of the IPSP exhibits distinct characteristics akin to those of the MCF. During the planning phase, the nodes in the port logistics network can be regarded as network nodes, the connections between logistics nodes can be expressed as network edges, and the various types of cargo in the port can be compared to different types of commodities in the network. Consequently, the operation plan generation problem in the planning phase can be conceptualised as the challenge of identifying the network path that maximises the loading and unloading requirements of vessels and trains in the dry bulk ports. Furthermore, various constraints in the IPSP reflect the attributes of the MCF, as detailed below.

- Network capacity constraint: The flow along edges and through nodes in the dry bulk port logistics network must not exceed their capacity, representing limitations imposed by logistics nodes such as equipment operation capacity, stockpile capacity, and train/vessel loading and unloading demands during actual operations. Variations in cargo demand and equipment capabilities also impact flow restrictions and transportation costs for different commodity flows in the MCF model.
- Flow balance constraint: The inflow and outflow at network nodes must be balanced to accurately reflect the interactions among multiple operational nodes within the operation path.
- Coal blending constraint: Dry bulk ports blend different raw coals/ores to meet diverse customer needs. This blending process constraint is integrated into the logistics network, ensuring that the flow from two incoming flows at the product coal/ore node adheres to specified ratio constraints.

As one of the classical problems in operations research, the resource allocation problem (Koopman, 1953) aims to minimise the cost function by assigning resources to participants, projects, or tasks. In the IPSP, this primarily involves the allocation of time, space, and physical resources. The diversity of resource allocation types and the dynamic availability of resources, such as inventory fluctuations during inbound and outbound operations, pose significant challenges to solving the IPSP.

- Stockpile capacity constraint: The allocation of stockpile resources must not exceed the stockpile inventory.

- Operation time constraint: The timing of inbound and outbound operations must adhere to the train's or vessel's scheduled time windows and their specified order, such as train inbound sequence, cabin loading/unloading sequence and ship loading/unloading batch sequence.
- Operation non-overlapping constraint: Due to the uniqueness of space or physical resources at any given time, operation tasks utilising the same space (e.g., stockpile) or physical resource (e.g., stacker) cannot overlap temporally.

The flow shop scheduling problem (Johnson, 1954), a classic sub-problem of the shop scheduling problem, is a widely used mathematical model in production management and scheduling, known for its NP-hard complexity. Treating operation tasks as processing jobs with different stages and operation equipment as processing machines allows for the abstraction of the IPSP scheduling phase into an ordered flow shop scheduling problem, as detailed in Lu et al. (2024). Additionally, some constraints in the scheduling phase share similarities with the ordered flow shop scheduling problem.

- Operation uniqueness constraint: Each operation task can be executed by only one operation stream, and each operation equipment can handle only one operation task at a time, similar to a job being processed by a single machine.
- Operation sequence constraint: Each device can only proceed to the next operation task after completing the current one. The operation tasks need to follow a specific process sequence.

The vehicle routing problem (VRP) is a classical NP-hard problem that deals with the efficient allocation of vehicles to multiple customer points, aiming to minimise total travel distance or costs while meeting various constraints (Golden et al., 2008). In the scheduling phase of the IPSP, there is a composite path planning problem involving various heterogeneous devices (e.g., stackers and reclaimers). The challenge is to efficiently allocate these devices to access the operational stockpile and complete operation tasks while considering the travel distance cost of the equipment. Some constraints in the IPSP are similar to those in the VRP, highlighting their similarities.

- Equipment moving distance constraint: After completing an operation task, the operating equipment must have enough time between consecutive tasks to move from its current operating point to the next operating point.
- Equipment operating capacity constraint: The amount of workload per unit of operation cannot exceed the operating capacity of the equipment unit. The unit operating capacity of the same equipment varies for different types of cargo.

In summary, the IPSP exhibits the combined properties of several NP-hard problems, including the MCF, resource allocation, flow shop scheduling, and VRP problems. This composite nature makes it challenging to describe and solve using a single-mode mathematical model. Moreover, the inclusion of production constraints, such as the equipment anti-collision constraint, further complicates the problem and presents additional challenges in finding solutions.

4. Hierarchical model for IPSP in a multi-cargo, multi-functional dry bulk port

Addressing the IPSP poses a significant challenge. The intricate nature of its combinatorial optimisation structure, along with the abundance of variables and constraints, complicates the task of formulating an integrated model for direct solver-based solutions. Appendix A presents the MIP model for IPSP, which includes numerous binary and integer variables capturing network flow and scheduling characteristics. As the problem scale increases, the difficulty of solving the model grows exponentially. To overcome these complexities, this study proposes a hierarchical approach. Specifically, distinct models are established for the planning and scheduling phases, incorporating two-phase interactions to enhance solution feasibility and efficiency. While integrated methods may enhance the verification of solution optimality, they often struggle with the

complexities of large-scale problems and are challenging to implement in ports accustomed to phased operations. The hierarchical solution framework proposed in this paper, aligning with the phased operational practices of ports, offers an efficient resolution of complex problems. Moreover, it proves to be more adaptable to the dynamic port operational environment, enabling rapid stage-wise planning or scheduling adjustments to accommodate fluctuations in port demand or supply.

4.1. A hierarchical solution framework to solve IPSP

The hierarchical solution framework outlined in this paper decomposes the IPSP into planning and scheduling phases, mirroring the manual decision-making process observed in dry bulk ports. Production planning and scheduling are tackled sequentially. As depicted in Fig. 5, during the planning phase, an MCNF model is constructed, accounting for workload constraints while relaxing scheduling constraints, and subsequently solved using a solver. The generated operation tasks and streams are then forwarded to the scheduling phase for the selection and scheduling of operation equipment and task timing. The following sections will elaborate on the implementation details within this framework.

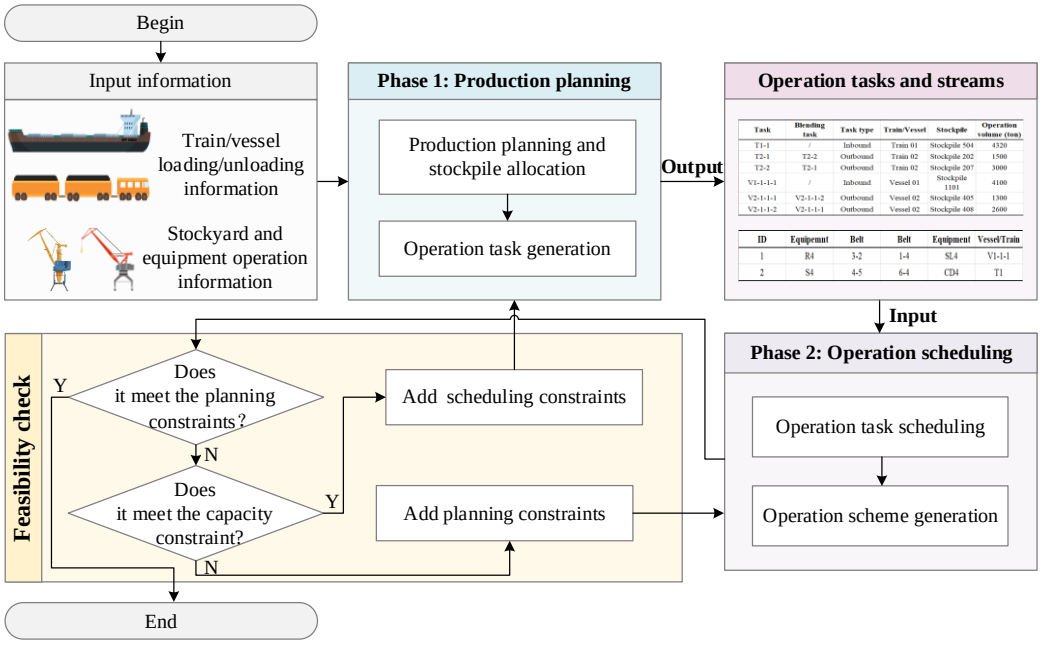


Fig. 5. A hierarchical solution framework.

To ensure consistency between the planning and scheduling phases, it is essential to maintain coherent information transmission and coordinated feasibility. The two-phase coupling in this study is achieved by transferring the operation tasks generated in the planning phase to the scheduling phase. After the scheduling process is completed, the resulting plan is verified against the planning constraints (constraints (4)–(9)), which include the demand-supply balance and task-allocation requirements of trains and vessels. If these constraints are satisfied, the IPSP is considered solved. If the plan violates the planning constraints, the algorithm then evaluates whether it satisfies the capacity constraint (constraint (13)), defined by the equipment's operational capability and stockpile capacity. When the capacity constraint is met, but the planning constraints are not, the scheduling constraints (constraints (28)–(33)) are incorporated into the planning phase to regenerate a logically consistent plan. Conversely, if the capacity constraint is violated, the planning constraints (constraints (4)–(9)) are transferred to the scheduling phase, enabling time-dependent sequencing and resource adjustments to resolve infeasibilities.

The planning constraints are checked first because they determine the logical feasibility of the operation plan, whether a valid operation task and cargo flow can exist at all. Only after confirming this logical feasibility does the algorithm verify the

physical feasibility through the capacity constraint. This hierarchical checking sequence is both conceptually consistent with the structure of the integrated problem and computationally efficient, as it avoids unnecessary resource-level evaluations for logically infeasible plans and effectively distinguishes whether infeasibility arises from unmet demand or limited operational capacity.

4.2. MCNF model for the planning phase

In the planning phase of IPSP, an MCNF problem is addressed to meet diverse customer requirements, as detailed in Section 3. The operation links within the dry bulk port, such as unloading/loading operations, stacking, reclaiming, and coal/ore blending, are abstracted as nodes across layers in the multi-commodity network. The interplay between adjacent operation links, such as the selection of stockpiles subsequent to unloading operations, is represented as edges within the network. Additionally, the volume of operations translates into flows within the network, thereby establishing the initial network diagram encompassing all potential edges. Given the multitude of trains/vessels with varying coal/ore requirements and the diverse raw ores/coals during blending operations, alongside the possibility of multiple trains or vessels occupying the same track or berth, respectively, multiple flows traverse multiple edges within the network. Consequently, the planning phase of the IPSP is effectively recast as an MCNF problem. It can be briefly defined as follows: Within a planning period T, all equipment and resources operate normally. Each train and vessel has specific loading and unloading requirements, as well as the latest departure time. The planning phase aims to fulfil the loading and unloading needs of all trains and vessels within the planning period, maximising customer satisfaction in the MCNF model. Table 3 presents the notations utilised in the model, where Latin letters and Greek letters denote parameters (indices and sets) and decision variables, respectively.

Table 3

Notations and descriptions for the planning phase of the IPSP.

Notation	Description
Set	
TR	Set of trains, $tr \in TR$, $TR = TR^+ \bigcup TR^-$, TR^+ denotes the set of trains with inbound operations, TR^- denotes the set of trains with outbound operations.
SP	Set of stockpiles, $sp \in SP$.
R	Set of rails, $r \in R$.
V	Set of vessels, $v \in V$, $V = V^+ \bigcup V^-$, V^+ denotes the set of vessels with inbound operations, V^- denotes the set of vessels with outbound operations.
$C(v)$	Set of cabins of vessel v, $c \in C(v)$.
$CB(v)$	Set of cabin batches of vessel v, $cb \in CB(v)$.
P	Set of cargo blending schemes, $p \in P$.
BP	Set of mixed product ore/coal, $bp \in BP$.
B	Set of berths, $b \in B$.
N	Set of all points in the multi-commodity network, $N = s \bigcup R \bigcup TR \bigcup SP \bigcup P \bigcup BP \bigcup CB \bigcup C \bigcup V \bigcup B \bigcup t$, where s and t denote the virtual source and sink nodes in the network, respectively.
A	Set of all edges in the multi-commodity network, $(i, j) \in A$, $i, j \in N$.
$Z^+(i)$	Set of incoming edges of node i, $i \in N \setminus \{s\}$.
$Z^-(i)$	Set of outgoing edges of node i, $i \in N \setminus \{t\}$.
Parameter	
n_p	Number of cargo types involved in blending scheme $p \in P$.
n_{bp}	Number of cargo types in mixed product ore/coal $bp \in BP$.
q_i	Number of mixed cargo types handled by train/cabin batch node $i \in TR^- \bigcup CB(v)$.
w_i	Number of single-product cargo types handled by train/cabin batch node $i \in TR^- \bigcup CB(v)$.
inv_{sp}	Initial inventory of stockpile sp.

cap_i	Maximum capacity of node $i \in N \setminus \{s,t\}$.
pro_{sp}^{p}	Proportion of cargo in stockpile sp in blending scheme p.
u_{i_j}	Maximum capacity of arc (i,j), determined by the operating capability or capacity of the connected nodes.
Variable	
φ_{i_j}	Flow on edge (i,j).
γ_{i_j}	Node connection variable, 1 if edge (i,j) is connected, 0 otherwise.

Based on the preceding description, the MCNF model in the planning phase of the IPSP is as follows:

$$\max \sum_{tr\in TR^{+}} \sum_{sp\in SP} \varphi_{tr\_sp} + \sum_{tr\in TR^{-}} \sum_{r\in R} \varphi_{tr\_r} + \sum_{cb\in CB^{+}} \sum_{sp\in SP} \varphi_{cb\_sp} + \sum_{v\in V^{-}} \sum_{b\in B} \varphi_{v\_b} \tag{1}$$

The objective function (1) aims to maximise customer demands within the multi-commodity network, specifically to maximise the loading and unloading requirements of trains and vessels. This ensures that the dry bulk port can meet these requirements with optimal resource utilisation, generating operation tasks for the scheduling phase to minimise the overall completion time. The constraints of the model can be categorised into the following five parts:

(1) Network balance constraints

$$\sum_{(i,j)\in Z^{+}(t)} \varphi_{i\_j} - \sum_{(i,j)\in Z^{-}(s)} \varphi_{i\_j} = 0 \tag{2}$$

$$\sum_{(i,j)\in Z^{-}(i)} \varphi_{i\_j} - \sum_{(j,i)\in Z^{+}(i)} \varphi_{j\_i} = 0 \qquad \forall i \in N \setminus \{s,t\} \tag{3}$$

Constraint (2) ensures that the total inflow equals the total outflow. The flow from the source point s represents the amount of inbound operations and existing stockyard inventory. The flow absorbed by the sink point t represents the amount of outbound operations and post-operation inventory. Constraint (3) ensures that each transfer point satisfies the condition where inflow equals outflow.

(2) Network capacity constraints

$$\sum_{k=1}^{n_p} \varphi_{sp\_p}^{k} \le u_{sp\_p} \qquad \forall sp \in SP, \forall p \in P \tag{4}$$

$$\sum_{k=1}^{n_{bp}} \varphi_{p\_bp}^{k} \le u_{p\_bp} \qquad \forall p \in P, \forall bp \in BP \tag{5}$$

$$\sum_{k=1}^{q_i} \varphi_{bp\_i}^{k} \le \sum_{k=1}^{q_i} u_{bp\_i}^{k} \qquad \forall bp \in BP, \forall i \in TR^{-} \bigcup CB(v), \forall v \in V^{-} \tag{6}$$

$$\sum_{k=1}^{w_i} \varphi_{sp\_i}^{k} \le \sum_{k=1}^{w_i} u_{sp\_i}^{k} \qquad \forall sp \in SP, \forall i \in TR^{-} \bigcup CB(v), \forall v \in V^{-} \tag{7}$$

$$\varphi_{v\_b} \le u_{v\_b} \qquad \forall v \in V^{-}, \forall b \in B \tag{8}$$

$$\varphi_{tr\_r} \le u_{tr\_r} \qquad \forall tr \in TR^{-}, \forall r \in R \tag{9}$$

$$\varphi_{s\_sp} = inv_{sp} \qquad \forall sp \in SP \tag{10}$$

$$0 \le \varphi_{i\_sp} + \varphi_{s\_sp} \le cap_{sp} \qquad \forall sp \in SP, \forall i \in TR^{+} \bigcup CB(v), \forall v \in V^{+} \tag{11}$$

Because a blending scheme may serve multiple cargo types, constraint (4) ensures that the total flow of all products along an edge from a stockpile to the blending scheme does not exceed the maximum capacity of that edge. Similarly, constraint (5) restricts the total flow of all products along an edge from the blending scheme to the product ore/coal, ensuring it remains within the edge's maximum capacity. For trains or vessels with multiple product ore/coal requirements, constraint (6) ensures that the inflow of product ore/coal entering a train or cabin batch node does not exceed the demand specified at that node. Constraint (7) applies to single-product cargoes other than product ore/coal, ensuring that the inflow into a train or cabin batch node does not surpass the demand limit for that cargo. Constraints (8) and (9) ensure that the flow from vessel node v to berth node b and from train node tr to rail node r does not exceed the total flow demand of vessel v or train tr, respectively. Constraint (10) represents the existing inventory at the stockpile node sp as the flow from the virtual starting node s to the

stockpile node sp. Constraint (11) guarantees that the sum of the flows to the stockpile node sp must satisfy the stockpile capacity constraint.

(3) Node connection constraints

$$\gamma_{i\_j} \in \{0,1\} \quad \forall (i,j) \in A \tag{12}$$

$$\sum_{k=1}^{q_j+w_j} \varphi_{i\_j}^{k} \le u_{i\_j} \times \gamma_{i\_j} \quad \forall (i,j) \in A \tag{13}$$

Constraint (12) defines the node connection variable. $\gamma_{i\_j}=1$ indicates that edge (i,j) is connected, establishing a relationship between node i and j. In this case, constraint (13) ensures that the sum of each cargo's flow on edge (i,j) is less than the edge capacity. Conversely, $\gamma_{i\_j}=0$ means that edge (i,j) is disconnected, with no connection between the two nodes, and the edge flow is zero.

(4) Disjoint path selection constraints

$$\sum_{sp\in SP} \gamma_{tr\_sp}=1 \quad \forall tr \in TR^{+} \tag{14}$$

$$\varphi_{tr\_sp} \le cap_{tr} \times \gamma_{tr\_sp} \quad \forall tr \in TR^{+}, \forall sp \in SP \tag{15}$$

$$\sum_{sp\in SP} \gamma_{cb\_sp}=1 \quad \forall cb \in CB(v), \forall v \in V^{+} \tag{16}$$

$$\varphi_{cb\_sp} \le cap_{cb} \times \gamma_{cb\_sp} \quad \forall cb \in CB(v), \forall v \in V^{+}, \forall sp \in SP \tag{17}$$

$$\sum_{bp\in BP} \gamma_{bp\_i}=1 \quad \forall i \in CB(v) \bigcup T^{-}, \forall v \in V^{-} \tag{18}$$

$$\varphi_{bp\_i} \le cap_{bp} \times \gamma_{bp\_i} \quad \forall bp \in BP, \forall i \in CB(v) \bigcup T^{-}, \forall v \in V^{-} \tag{19}$$

Trains with inbound operations typically have multiple optional unloading stockpiles; however, due to production constraints, each train can only select one stockpile for unloading. This is reflected in the multi-commodity flow model, where only one edge connects the train node tr to the stockpile node sp, as specified by constraint (14), and the flow on this edge cannot exceed the edge capacity, as specified by constraint (15). Similarly, constraints (16) and (17) ensure that only one edge connects the cabin batch node cb to the stockpile node sp, with the edge flow not exceeding its capacity. Constraints (18) and (19) guarantee that only one edge connects the product ore/coal node to the train or cabin batch node, while also restricting the edge flow to its capacity.

(5) Coal blending ratio constraint

$$\varphi_{s1\_p} / \varphi_{s2\_p} = pro_{s1}^{p} / pro_{s2}^{p} \quad \forall s1, s2 \in SP(p), \forall p \in P \tag{20}$$

Constraint (20) is the edge flow ratio constraint, ensuring that the edge flows belonging to the same blending scheme p meet a specified blending ratio. This allows for the proper mixing of two raw coals/ores into a product coal/ore.

4.3. CP model for the scheduling phase

In the planning phase of the IPSP, the demand of each train or vessel, indexed by i, is decomposed into a set of operation tasks indexed by interval variable α, where $\alpha \in OT(i)$. The scheduling phase takes these generated operation tasks as input parameters. A scheduling model is then formulated to account for multiple operational constraints. This paper presents the model using mathematical notation, incorporating nonlinear elements like logical operators, variable subscripts, and specialised constraints in CP, such as cumulative and non-overlap constraints. The concept of CP was first proposed in the fields of artificial intelligence and graphics during the 1960s and 1970s, and has since been applied and studied across various computing fields. CP has proven to be suitable for solving many combinatorial optimisation problems that can be expressed in different forms, especially for scheduling problems (Müller et al., 2022). Compared to the MIP model, the CP model can reduce computational complexity by reducing the number of variables and constraints. For example, in the job shop scheduling problem with m

machines and n jobs, the MIP model requires mn variables and $m+n$ constraints, whereas the CP model can represent the same problem with only n variables and a single non-overlap constraint.

An important fundamental concept in the CP model is the interval variable. Interval variables have attributes such as presence, start, end, and length. They are used to model quantities with interval characteristics, and their unique existence attributes make their presence optional (when the presence status is absent, it indicates that the variable is not included in the solution). The CP language provides numerous functions related to interval variables, most of which have intuitive and concise meanings. For example, $start(a)$ represents the starting point of interval variable a.

The model is linearised and solved by utilising the CP Optimizer (Laborie et al., 2018). It can be briefly described as follows: During a planning period T, all equipment and resources are available for use. The operation tasks are generated in the planning phase. The objective of the scheduling phase is to complete all operation tasks as efficiently as possible within the planning period while adhering to production constraints and the latest departure times of trains and vessels. Table 4 presents the notations utilised in the CP model during the scheduling phase, while some parameter and set definitions are provided in Table 3.

Table 4

Notations and descriptions for the scheduling phase of the IPSP.

Notation	Description
Set	
OT	Set of operation tasks in the scheduling period, $\alpha \in OT$, $OT = OT^+ \bigcup OT^-$, OT^+ denotes the set of inbound operation tasks, OT^- denotes the set of outbound operation tasks.
CA	Set of cargo types, $ca \in CA$.
DS	Set of dump stations, $ds \in DS$.
GC	Set of gantry cranes, $gc \in GC$.
SU	Set of ship unloaders, $su \in SU$.
BL	Set of belt lines, $bl \in BL$.
ST	Set of stackers, $st \in ST$.
RC	Set of reclaimers, $rc \in RC$.
SR	Set of stacker-reclaimers, $sr \in SR$.
SL	Set of ship loaders, $sl \in SL$.
LS	Set of loading stations, $ls \in LS$.
M	Set of all production resources on the stockyard, $M = R \bigcup DS \bigcup GC \bigcup SU \bigcup BL \bigcup SP \bigcup ST \bigcup RC \bigcup SR \bigcup SL \bigcup LS \bigcup B$.
$OS(\alpha)$	Set of operation streams of operation task α, $\alpha \in OT$.
T	Set of time steps in the planning and scheduling period in minutes, $\tau \in \{1, \cdots, \lvert T \rvert\}$.
G	Set of yard equipment tracks, $g \in G$.
$M(g)$	Set of production resources on the same track g, $g \in G$.
PR	Set of production resource types, $PR = \{DS, GC, SU, BL, ST, RC, SR, SL, LS\}$.
$M(pr)$	Set of production resources of the same type pr, $pr \in PR$.
Parameter	
siz_α	The operation volume of operation task α.
oe_m^{ca}	Operating efficiency of equipment m handling cargo ca.
est_α	The earliest start time of operation task α, which is determined by the arrival time of the train/vessel associated with operation task α.
let_α	The latest end time of operation task α, which is determined by the departure time of the train/vessel associated with operation task α.
$lead$	The operation task switching time of the equipment.
ord_α	The operation sequence of operation task α, which is determined by the train/vessel arrival sequence, vessel loading batch and cabin loading/unloading sequence, etc.
cap_m	Maximum capacity of resource m.
pos_β	The operation position of operation stream β, which is determined by the operation stockpile or the operation cabin.
spd_m	Unit moving speed of equipment m in m/min.
saf_m	The safe distance that equipment m needs to keep from other equipment.

let_{tr}	The latest operation end time of train $tr \in TR$, which is determined by the departure time of the train.
let_{v}	The latest operation end time of vessel $v \in V$, which is determined by the departure time of the vessel.
Variable	
α	Interval variable of an operation task.
β	Optional interval variable of an operation stream.
δ_m	Cumulative function of resource m.
$\chi(\beta)$	Indicator of operation stream β in the sequence interval variable list.

Based on the preceding description and the given operation task list, the CP model in the scheduling phase of the IPSP is as follows:

$$\min \max\left\{end(\alpha)\middle|\alpha \in OT\right\} \tag{21}$$

The objective function (21) aims to minimise the latest completion time of all operation tasks. This ensures that the time trains and vessels spend in port is minimised, fulfilling the needs of port production management. The model's constraints can be divided into the following five parts:

(1) Operation task constraints

$$sizeOf(\alpha) \geq siz_{\alpha} \Big/ min\left\{oe_m^{ca(\alpha)}\middle|m \in M(\alpha) \setminus \{R, SP, B\}\right\} \quad \forall \alpha \in OT \tag{22}$$

$$alternative(\alpha, OS(\alpha)) \quad \forall \alpha \in OT \tag{23}$$

$$est_{\alpha} \leq start(\alpha) < let_{\alpha} \quad \forall \alpha \in OT \tag{24}$$

$$est_{\alpha} \leq start(\beta) < let_{\alpha} \quad \forall \beta \in OS(\alpha), \forall \alpha \in OT \tag{25}$$

$$endBeforeStart(\alpha, \alpha', lead) \quad \left\{(\alpha, \alpha')\middle|ord_{\alpha} < ord_{\alpha'}, \forall \alpha, \alpha' \in OT\right\} \tag{26}$$

$$noOverlap(sequence(OT(m))) \quad \forall m \in M \tag{27}$$

Operation task constraints include operation stream selection and operation time constraints. The *sizeOf* constraint in constraint (22) ensures that the length of the operation task is sufficient to complete operation task α, and the operating efficiency of equipment m is related to cargo type ca. The *alternative* constraint in constraint (23) guarantees that exactly one operation stream $\beta \in OS(\alpha)$ completes operation task α. Constraints (24) and (25) ensure that the start time of the operation task α and the operation stream β are not earlier than the arrival time of the train/vessel and not later than their latest departure time. The *endBeforeStart* constraint in constraint (26) restricts the subsequent operation task α' to start after the end of the previous operation task α, ensuring that the operation sequence adheres to production requirements such as train arrival sequence, vessel berthing sequence, loading batch and cabin loading sequence. The *noOverlap* constraint in constraint (27) ensures that the set of operation tasks $OT(m)$ arranged by start time maintains non-overlapping with each other, preventing operation time overlap for tasks using the same production resource m.

(2) Stockpile operational constraints

$$\delta_{sp} = inv_{sp} + \sum_{\alpha \in OT^{+}(sp)} stepAtStart(\alpha, siz_{\alpha}) - \sum_{\alpha \in OT^{-}(sp)} stepAtStart(\alpha, siz_{\alpha}) \quad \forall sp \in SP \tag{28}$$

$$alwaysIn(\delta_{sp}, 0, |T|, 0, cap_{sp}) \quad \forall sp \in SP \tag{29}$$

The bulk port stockyard is the storage place for various bulk cargoes, serving as the intersection and buffer for inbound and outbound operations. Through the cumulative function *stepAtStart* in constraint (28), the cumulative resource δ_{sp} increases by the amount of siz_{α} at the beginning of α, ensuring the inventory of stockpile sp increases with inbound operations and decreases with outbound operations. Constraint (29) ensures that the stockpile inventory remains non-negative and does not exceed the maximum stockpile capacity during the scheduling period $[0, |T|)$.

(3) Equipment operational constraints

$$noOverlap(sequence(OS(m)), TM) \quad \forall m \in GC \cup SU \cup ST \cup RC \cup SR \tag{30}$$

$$TM[\chi(\beta 1), \chi(\beta 2)] = |pos_{\beta 1} - pos_{\beta 2}| \Big/ spd_m \quad \forall \beta 1, \beta 2 \in OS(m) \tag{31}$$

$$noOverlap\left(sequence\left(\beta 1,\beta 2\right),TM\right)\quad \forall \beta 1\in OS\left(m1\right),\forall \beta 2\in OS\left(m2\right),\forall m1,m2\in M\left(g\right),\forall g\in G \tag{32}$$

$$TM\left[\chi\left(\beta 1\right),\chi\left(\beta 2\right)\right]=\left[\left|pos_{\beta 1}-pos_{\beta 2}\right|+max\left(saf_{m1},saf_{m2}\right)\right]/min\left(spd_{m1},spd_{m2}\right)$$
$$\forall \beta 1\in OS\left(m1\right),\forall \beta 2\in OS\left(m2\right),\forall m1,m2\in M\left(g\right),\forall g\in G \tag{33}$$

Eqs. (30)–(33) simulate the operation process of dry bulk port equipment. Constraint (30) ensures that a device m performing continuous operation tasks has sufficient time to move from the previous position to the next. Eq. (31) provides the calculation for the required moving time between the operating locations of adjacent operation streams on device m. Constraint (32) ensures that two devices operating on the same track maintain a safe distance, preventing collisions or device crossing. Eq. (33) provides the calculation of the required moving time between operating locations for devices sharing the same track.

(4) Special operational constraints

$$start\left(\beta_1\right)=start\left(\beta_2\right)\quad \forall sp\left(\beta_1\right),sp\left(\beta_2\right)\in p,\forall p\in P \tag{34}$$

$$max\left\{end\left(\alpha\right)\middle|\alpha\in OT\left(tr\right)\right\}\le let_{tr}\quad \forall tr\in TR \tag{35}$$

$$max\left\{end\left(\alpha\right)\middle|\alpha\in OT\left(v\right)\right\}\le let_{v}\quad \forall v\in V \tag{36}$$

Constraints (34)–(36) address specific requirements arising from the production process of dry bulk ports. Constraint (34) ensures that operation streams belonging to the same blending scheme start simultaneously, enabling the even blending of two raw coals/ores into the product coal/ore required by the customer. Constraints (35) and (36) ensure that all train/vessel operation tasks are completed before the train's/vessel's latest departure times, thereby avoiding any delay costs.

(5) Redundant constraints

$$\delta_{pr}=\sum_{m\in M\left(pr\right)}\sum_{\beta\in OS\left(m\right)}pulse\left(\beta,1\right)\quad \forall pr\in PR \tag{37}$$

$$alwaysIn\left(\delta_{pr},0,\left|T\right|,0,cap_{pr}\right)\quad \forall pr\in PR \tag{38}$$

The use of constraint recombination, global constraints, redundant constraints, and surrogate constraints can significantly enhance the effectiveness of the solution process (Lazaar et al., 2012). Given that the problems in this study involve numerous resource allocation links, redundant resource constraints (37)–(38) are designed. These constraints limit the cumulative occupation of pr -type production resources over a specific time interval to ensure it does not exceed the total number of such devices available. These redundant constraints are strategically utilised to accelerate the convergence of the CP model. Adding or deleting these constraints does not affect the generation of optimal or feasible solutions for the model.

5. Solution methods

Since the MIP model of the IPSP contains numerous integer variables and quadratic constraints, solving the model is very time-consuming. Even for small-scale instances with only one train and one ship, the commercial solver CPLEX cannot find a feasible solution in a reasonable time. Considering that the IPSP problem has a decomposable structure and is usually decomposed into two stages, planning and scheduling, in the process of bulk port management, this paper proposes a hierarchical solution framework based on MCNF and CP, as illustrated in Fig. 6.

To address the limitations of CP solvers, namely their tendency to fall into local optima and reduced scalability, this paper introduces a dual-engine optimisation method combining DQN (Mnih et al., 2015) and CP, embedded within a hierarchical solution framework. The method leverages the precise solution capabilities and feasibility guarantees of CP, alongside DQN's capacity for real-time decision-making based on historical data. This hybrid approach mitigates the limitations of CP's restricted global search and DRL's scalability and training costs in large-scale problems.

The proposed method proceeds as follows: After initialising the problem instance, the CPLEX solver generates the operation tasks in the planning phase, followed by the CP solver for task scheduling. During scheduling, a dual-driven optimisation

strategy combining the CP solver and a DQN agent is applied. If the CP solver converges within the specified time limit, the solution is returned. Otherwise, the DQN agent is invoked to guide decision updates. The DQN agent selects appropriate operation streams for operation tasks to assist the CP solver. These inputs help the solver escape local optima and explore improved solutions. The best solution identified during the search process is ultimately returned.

The structure of this section is as follows: Section 5.1 presents the hierarchical solution algorithm based on MCNF and CP, while Section 5.2 details the design of the DQN algorithm and the interaction framework between the CP and DQN algorithms.

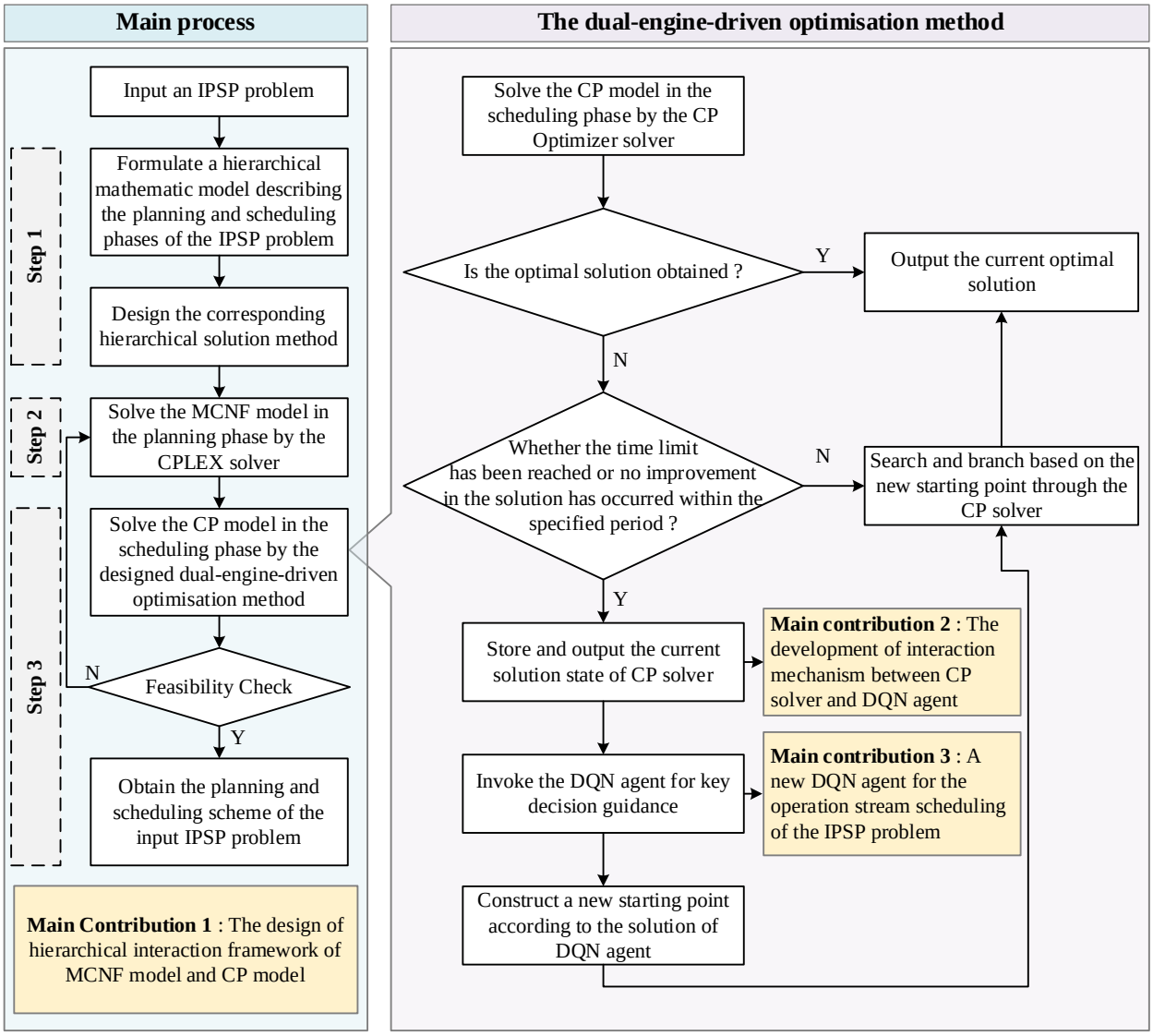


Fig. 6. The process of the proposed optimisation methods.

5.1. A hierarchical approach based on MCNF and CP

In this section, a hierarchical solution method based on MCNF and CP is presented, with the solution steps outlined in Algorithm 1. The MCNF model is modelled and solved using the IBM ILOG CPLEX solver, while the CP model is modelled and solved using the IBM ILOG CP Optimizer solver. First, the MCNF model is solved by the simplex optimiser of the CPLEX solver to achieve maximum flow distribution in the dry bulk port logistics network, meeting the loading and unloading requirements of trains and vessels. Based on the flow distribution of each edge, train and vessel operation tasks are generated. The CP model based on operation tasks is then solved using the search algorithm and constraint propagation process of the CP

Optimizer to minimise the completion time of the operation tasks. If the CP Optimizer detects inconsistencies during the predefined constraint propagation process, it will backtrack until a feasible solution is found. After solving the CP model, the overall solution is verified through a hierarchical constraint check. When infeasibilities are detected, relevant constraints are iteratively fed back between the planning and scheduling phases to ensure both logical consistency and resource feasibility of the integrated solution. Specifically, the solution is first verified against the planning constraints (constraints (4)–(9)) and subsequently evaluated with respect to the capacity constraint (constraint (13)). If the capacity constraint is violated, the planning constraints (constraints (4)–(9)) are incorporated into the CP model in the scheduling phase to regenerate a feasible solution. Conversely, if the capacity constraint is met but the planning constraints are violated, the scheduling constraints (constraints (28)–(33)) are added to the MCNF model in the planning phase to produce a revised and feasible plan.

Algorithm 1: Hierarchical solution method based on MCNF and CP

Input: Port layout, stockpile and equipment information, instance parameters, solver parameters
Output: production planning and scheduling schemes
Initialise basis for the planning phase of IPSP and set CPLEX solver parameters
while the optimal solution s_1 is not found *or* the problem is unbounded **do**
  Solve the linear relaxed problem using the simplex method
  Calculate reduced costs σ for all non-basic variables
  if all the σ are non-negative **then**
    Output optimal solution s_1 and break
  end if
  Select the non-basic variable x_i with the most negative reduced cost σ_i
  Calculate the direction vector and step length
  Update variable values and basis
  if the current solution s_1' is in integral form **then**
    Output optimal integer solution $s_1 = s_1'$ and break
  else
    Apply multi-commodity flow cuts to the problem
  end if
end while
Generate operation tasks based on the planning solution
Set CP Optimizer solver parameters and initial constraint propagation
while the search tree is not fully traversed **do**
  Perform search and update the solution if it is improved
  if a variable $\alpha = \varnothing$ **then**
    Terminate search and output no feasible solution
  end if
end while
Output optimal scheduling phase solution s_2'
if the solution satisfies the planning constraints **then**
  Terminate and output the final optimal solution $s_2 = s_2'$
else if the solution s_2' satisfies the capacity constraint **then**
  Add scheduling constraints to the MCNF model and return to step 3
else
  Add planning constraints to the CP model and return to step 21
End if

The CPLEX and CP solvers used in this paper offer various solution parameters and strategy settings that can significantly affect their performance and efficiency. For example, the multi-commodity flow (MCF) cuts parameter in the CPLEX solver can be set to add multi-commodity flow cuts to accelerate the solution of the MCNF model. The presolve function of the CP Optimizer simplifies the CP model by reducing linear constraints, eliminating redundancies, and removing fixed expressions before initiating the search. This process enhances computational efficiency and improves the overall model solution. Detailed

descriptions of these parameters and strategy settings can be found in the solver documentation and related literature (Laborie et al., 2018). The solver parameter settings used in this paper are shown in 错误!书签自引用无效。. By setting these parameters correctly, hundreds of redundant expressions can be effectively reduced, and the branch or search speed can be accelerated.

Table 5

Solver parameters and strategy settings.

Solver	Parameter	Value/Search strategy
CPLEX	Presolve	Turn on
	MIP flow cover cuts	Do not generate flow cover cuts
	Multi-commodity flow cuts	Generate a moderate number of MCF cuts
	Repeat presolve	Turn off
CP Optimizer	Presolve	Turn on
	Cumulative Function inference level	Default
	Sequence inference level	Extended
	Precedence inference level	Medium
	NoOverlap inference level	Extended
	Workers	6

5.2. A dual-engine-driven optimisation method based on CP and DQN

Relaxation bounds and relevant improvement mechanisms are central algorithms in current commercial solvers (Cappart et al., 2022). Given the limitations of solvers in achieving optimal solutions and handling large-scale problems, scholars have explored heuristic algorithms to assist in the solvers' decision-making process. However, these heuristic approaches are often time-consuming and lack mathematical rigour and optimality proofs (Bengio et al., 2021). On the other hand, DRL has recently shown promising application prospects in designing heuristic algorithms for solving combinatorial optimisation problems (Cappart et al., 2021). This section presents a dual-engine-driven method that integrates a DQN agent with a CP solver. By combining these methods, this approach leverages the backtracking and pruning mechanisms of CP to iteratively enhance the solution quality, while using a pre-trained DQN agent to guide key variable decisions. This integration not only reduces the computational load on the CP solver but also significantly improves the efficiency of the solution process. The following subsections outline the design of the DQN approach, as illustrated in Fig. 7, and describe the development of an interactive framework that integrates the DQN agent with the CP solver.

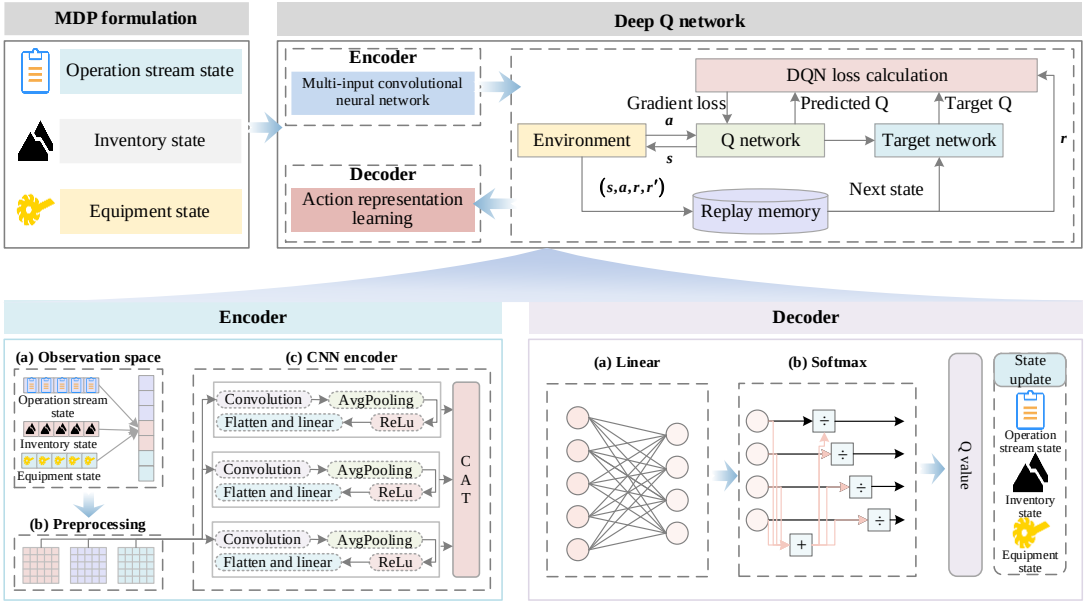


Fig. 7. The workflow and design of the DQN approach.

5.2.1. Markov decision process formulation

The solution process of the scheduling phase of the IPSP can be naturally described as a Markov decision process (MDP), where the agent realises operation task scheduling, operating stockpile and equipment selection by selecting the operating stream. The MDP can be characterised by the tuple $(\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{\gamma})$, with each element defined as follows. Notably, to distinguish from the MIP and CP models, notations related to the MDP and the DQN agent in this paper are marked in bold.

The state space $\boldsymbol{S}$ encompasses all possible states of all operation streams, equipment and stockpile inventory levels. It is described through three different channels:

(1) Operation stream state $\boldsymbol{S^{OS}}$

This channel retrieves the status of all operation stream variables. It is defined as $\boldsymbol{S^{OS}} = \left\{\boldsymbol{s_i^{OS}}\right\}_{\boldsymbol{i}\in\{1,\cdots,\boldsymbol{n}\}}$, where $\boldsymbol{s_i^{OS}} = \left(\boldsymbol{i}, \boldsymbol{o_i}, \boldsymbol{d_i}, \boldsymbol{ty_i}, \boldsymbol{vm_i}, \boldsymbol{sp_i}, \boldsymbol{q_i^1}, \boldsymbol{q_i^2}, \boldsymbol{st_i}, \boldsymbol{et_i}\right)$, $\boldsymbol{i}$ denotes the operation stream index, $\boldsymbol{o_i}$ represents the index of the operation task that operation stream $\boldsymbol{i}$ belongs to, $\boldsymbol{d_i}$ indicates whether the operation task is scheduled, $\boldsymbol{ty_i}$ specifies the task type (1 for inbound tasks and 2 for outbound tasks), $\boldsymbol{vm_i}$ denotes the task volume, $\boldsymbol{sp_i}$ is the operational stockpile for operation stream $\boldsymbol{i}$, $\boldsymbol{q_i^1}$ is the handling equipment for operation stream $\boldsymbol{i}$ (including dump stations, loading stations, ship loaders, ship unloaders, and gantry cranes), $\boldsymbol{q_i^2}$ is the yard stacking/reclaiming equipment for operation stream $\boldsymbol{i}$ (including stackers, reclaimers and stacker-reclaimers), $\boldsymbol{st_i}$ is the start time, and $\boldsymbol{et_i}$ is the end time.

(2) Inventory state $\boldsymbol{S^{SP}}$

This channel retrieves the status of all stockpile inventory. It is defined as $\boldsymbol{S^{SP}} = \left\{\boldsymbol{s_j^{SP}}\right\}_{\boldsymbol{j}\in\{1,\cdots,\boldsymbol{m}\}}$, where $\boldsymbol{s_j^{SP}} = \left(\boldsymbol{t_j}, \boldsymbol{f_j}, \boldsymbol{u_j}\right)$, $\boldsymbol{t_j}$ is the latest operation completion time of stockpile $\boldsymbol{j}$, $\boldsymbol{f_j}$ is the current inventory quantity of stockpile $\boldsymbol{j}$, and $\boldsymbol{u_j}$ is the maximum inventory quantity of stockpile $\boldsymbol{j}$.

(3) Equipment state $\boldsymbol{S^{EQ}}$

This channel retrieves the status of two types of equipment (handling equipment and stacking/reclaiming equipment) at each step. It is defined as $\boldsymbol{S_q^{EQ}} = \left\{\boldsymbol{s_q^{EQ}}\right\}_{\boldsymbol{q}\in\{1,\cdots,\boldsymbol{q}1,\boldsymbol{q}1+1,\cdots,\boldsymbol{q}1+\boldsymbol{q}2\}}$, where $\boldsymbol{q}1$ represents the index of handling equipment and $\boldsymbol{q}2$ represents the index of stacking/reclaiming equipment. The $\boldsymbol{s_q^{EQ}} = \left(\boldsymbol{t_q}\right)$ indicates the latest operation completion time of equipment $\boldsymbol{q}$.

The action set $\boldsymbol{A}$ consists of the operation stream indices selected in sequence. By reading the operation stream index, the stockpile index and equipment index that make up the operation stream can be identified, thereby completing the operation stockpile and equipment selection, as well as the scheduling sequence of the operation tasks.

The transition $\boldsymbol{P}$ refers to a deterministic rule where the next state is derived based on the current state and the action taken. The transition in this paper is achieved by updating the stockpile inventory, the latest completion time of the stockpile, and the equipment according to the selected operation stream.

Considering the intricate structure of the IPSP, the reward function is designed using a staged reward mechanism with reference to Xu et al. (2024). Intermediate rewards are provided when the agent successfully selects a correct operation stream, encouraging incremental progress towards the target. The current reward is determined based on the completion time of the operation task. Upon completing all operation task scheduling, a larger terminal reward is granted. Conversely, if the agent's decision violates constraints such as stockpile capacity limits or the availability and uniqueness of operation streams, a penalty is imposed, and the optimisation process is terminated. The reward $\boldsymbol{R}$ is defined as follows:

$$\boldsymbol{R}^1\left(\boldsymbol{s},\boldsymbol{a}\right) = \begin{cases} -1000 & \text{violate constraints} \\ 1000 & \text{meet constraints} \\ -100\times\left(\boldsymbol{et_a} - \boldsymbol{C}\right) & \text{Complete the scheduling of a task} \\ 100000 & \text{Complete all tasks} \end{cases} \tag{39}$$

$$\boldsymbol{R} = \sum \boldsymbol{R}^1\left(\boldsymbol{s},\boldsymbol{a}\right) \tag{40}$$

where $C = max_{i \in \{1,\cdots,n\}} et_i$ is the latest completion time of all scheduled operation tasks at each step.

The discount factor γ is set to 0.9 and serves as a parameter to balance immediate and future rewards (Xu et al., 2024). The goal of this MDP is to learn a policy π_θ, which determines the optimal action for the agent at each state, maximising the expected cumulative rewards while minimising the maximum completion time of all operation tasks.

5.2.2. Encoder and decoder

As outlined in Section 4, the state information during the IPSP scheduling phase encompasses all operation stream details, inventory levels, and equipment statuses. To maximise rewards in operation scheduling, the encoded state is structured to systematically represent global information, including the statuses of operation streams, inventories, and equipment. Each input type is processed through a dedicated branch, resulting in feature tensors. To address input heterogeneity, a multi-input convolutional neural network (CNN) architecture (Lecun et al., 1998) is employed to extract meaningful feature representations from each input type. By concatenating these inputs, the model achieves efficient multi-dimensional feature extraction and processing. The structure of the CNN encoder for the operation stream state branch is illustrated in Fig. 8, and the encoders for the inventory and equipment states have similar structures.

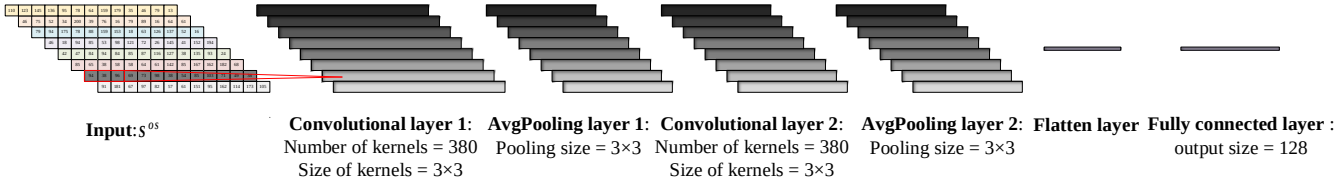


Fig. 8. The structure of the CNN encoder for the operation stream state.

The specific process is as follows:

(1) Concatenate the state features of operation streams, inventory levels, and equipment statuses into separate vectors;

(2) Normalise the state values to a range of $[0,1]$ to ensure uniform scaling across features;

(3) Transform the different state vectors into a fixed-size input embedding through multiple hidden layers;

(4) Combine the state vectors into a unified one-dimensional input embedding.

The CNN encoder is specifically designed to process these inputs efficiently. As shown in Fig. 8, the operation stream operates independently, with no connections to other streams, thereby treating each stream as a separate channel. Padding is utilised within each channel to facilitate the convolutional calculation. The initial convolutional layer employs 3×3 kernels to process the input matrix. An average pooling (avgPooling) layer follows, reducing the spatial dimensions of the data and thereby decreasing both the number of parameters and computational complexity. The second convolutional layer employs 3×3 kernels for further encoding, followed by another avgPooling layer. Finally, a flatten layer converts the encoded state embeddings into a one-dimensional representation.

The primary role of the decoder is to convert the feature vector generated by the encoder into Q-values for action decision-making. The decoder design is outlined as follows:

(1) State feature extraction: Compute model parameters and biases using the state feature vector;

(2) Softmax processing and Q-value output: Apply the Softmax function to produce Q-values distributed across the action space. Actions are selected based on the maximum Q-value, determining the chosen operation stream.

5.2.3. Training

To improve the convergence of the decoder in the scheduling phase of the IPSP, techniques such as experience replay, target networks for stability, and action representation learning are employed to efficiently manage the large action space. The training process consists of the following components:

(1) Learning with DQN

The DQN framework iteratively updates the Q-network using experiences collected from the environment. During each training step:

1) The state is encoded into features that represent operation streams, stockpiles and equipment;

2) The Q-network predicts Q-values for all possible actions in the action space;

3) The agent selects actions using an ε-greedy policy to balance exploration and exploitation;

The Q-values are updated using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a} Q(s',a') - Q(s,a)\right] \tag{41}$$

The loss function minimises the temporal difference (TD) error:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')} \sim D\left[\left(r + \gamma \max_{a'} Q(s',a';\theta') - Q(s,a;\theta)\right)^2\right] \tag{42}$$

where D is the replay buffer, θ represents the Q-network parameter, and θ' is the target network parameter.

The approximation error of the Q-value function influences the agent's estimation of future rewards and thus its action selection. During training, the Q-network is updated to minimise the discrepancy between its current prediction and the target Q-value. When this discrepancy, known as the Bellman residual, is small, the learned Q-function provides a near-accurate estimate of the optimal cumulative reward. Recent theoretical studies suggest that under the assumption of sufficient network capacity, this approximation error remains bounded (Fan et al., 2020).

In the proposed dual-engine framework, the DQN component determines only the key decision variables, while the CP solver reoptimises the remaining variables to ensure feasibility and local optimality. This division confines the influence of Q-value approximation errors to a limited portion of the decision space. Even when the predicted Q-values deviate slightly from their true values, the CP module mitigates these errors by refining the downstream optimisation process. The convergence analysis and algorithmic comparisons in Section 6.3 further demonstrate that the final solutions remain close to the solver's lower bound, confirming that the effect of Q-value approximation error on overall scheduling performance is minimal and effectively controlled.

(2) Learning with action representation

To address the large action space, action representation learning is employed to learn an embedding for actions, improving sampling efficiency:

1) Actions are embedded into a fixed-size latent space using an encoder-decoder structure;

2) A supervised learning approach predicts action embeddings by minimising the cross-entropy loss:

$$L_{supervised} = -\sum_{a} y_a \log \hat{y}_a \tag{43}$$

where y_a is the ground truth action distribution, and $\hat{y}_a$ is the predicted distribution. The embeddings are used to sample and refine actions.

(3) Algorithm

The overall learning process is summarised in Algorithm 2. In brief, this algorithm optimises the policy through reinforcement learning while enhancing convergence and efficiency by integrating action representation learning via supervised learning. This dual approach significantly improves the DQN agent's performance in solving the scheduling phase of the IPSP.

Algorithm 2: Learning process of the DQN agent

Input: training instances with n operation streams
Output: parameter θ of the policy
Initialise action representation parameter ϕ
Initialise Q-network parameter θ and target network parameter θ'
Initialise replay buffer D
for $episode = 1, 2, 3, \cdots, MaxEpisodes$ **do**
  Initialise state s from the training set
  for $t = 0, 1, \cdots, MaxSteps$ **do**
    Select an action a using ε-greedy policy with Q-values $Q(s, a; \theta)$
    Execute a, observe reward r and next state s'
    Store (s, a, r, s') in replay buffer D
    Sample a mini-batch of experiences (s, a, r, s') from D
    Compute target Q-values $y = r + \gamma \max_{a'} Q(s', a'; \theta')$
    Update Q-network θ by minimising $L(\theta)$
    Periodically update target network: $\theta' \leftarrow \theta$
    Update action representation ϕ using supervised learning loss $L_{supervised}$
  end for
end for

5.2.4. Algorithm structure and complexity analysis

This section introduces the interaction mechanism of the proposed dual-engine-driven optimisation method based on DQN and CP, followed by an analysis of its algorithmic complexity.

(1) Synergistic coupling between DQN and CP

This study proposes a dual-engine framework integrating DQN and CP, inspired by the work of Cappart et al. (2021). Unlike the existing approach, where the DQN agent directly guides the branching decisions within the CP solver, the proposed method exploits the inherent structure of the IPSP. Specifically, the DQN agent is trained to identify and fix key decision variables that exert the greatest influence on the CP solver's branching process, thereby reducing infeasible search paths and accelerating convergence.

A dedicated training environment is developed based on the CP model, enabling the DQN agent to learn the mapping between the solver's current state and the selection of effective decision variables. During the solution process, the trained agent observes the current state and determines the optimal operation stream for each operation task according to its learned policy. The CP solver then fixes these DQN-selected variables and optimises the remaining time-dependent variables. The interaction between the CP solver and the DQN module is implemented through the solver's IloSolution class. The allocation of decision variables between the DQN and CP modules is summarised in Table 6.

Table 6

Variable allocation between the DQN agent and the CP solver.

Variable	Description	Decision maker	Rationale
$presence(\beta)$	State variable of operation stream β, representing whether β is selected; determines the assignment between operation tasks and operation streams.	DQN agent	High-dimensional combinatorial decision with weak temporal coupling, suitable for the DRL method.
$start(\beta)$, $end(\beta)$, $start(\alpha)$, $end(\alpha)$	Start and end times of interval variables.	CP solver	Large-domain discretised time variables that can be efficiently handled through constraint propagation and precedence reasoning.
δ_m	Resource utilisation indicator of resource m.	CP solver	Determined by global constraints and cumulative functions of the CP solver to ensure resource feasibility.

The separation of decision variables between DQN and CP is guided by both computational and theoretical considerations. The DQN module is responsible for learning high-dimensional discrete policies (Mnih et al., 2015), whereas the CP solver focuses on temporal inference and constraint propagation (Laborie et al., 2018). Allowing the DQN agent to fix key variables effectively reduces the branching nodes of the CP search tree, thereby enhancing computational efficiency (Cappart et al., 2021). Empirical analysis of solver logs further reveals that determining the appropriate operation stream consumes substantial computational time, and suboptimal branching choices can drastically increase runtime. Prior studies in solver optimisation similarly indicate that variable selection often dominates computational cost, and poor branching decisions can lead to exponential growth in solution time (Jiang et al., 2023).

Moreover, due to the extremely large and interdependent decision space of the IPSP, directly applying DQN to solve the full combinatorial problem leads to convergence instability. As demonstrated in Table 10, the proposed DQN-CP framework not only achieves higher solution quality but also significantly reduces computation time compared with standalone CP approaches, confirming the effectiveness and rationality of the proposed variable allocation strategy. Through this collaborative mechanism, the DQN component effectively reduces the search space of the CP solver, while CP guarantees feasibility and local optimality within the reduced subspace.

(2) Algorithm complexity analysis

The computational cost of the proposed hierarchical framework mainly arises from three components: the MCNF model in the planning phase (solved by CPLEX), the CP model in the scheduling phase (solved by CP Optimizer), and the DQN-based decision module used to fix key variables. The convergence and optimisation capability of the overall framework are determined by these three components. Regarding the solvers, CPLEX employs Branch-and-Cut while CP Optimizer utilises Branch-and-Propagate. Both are exact methods that, given sufficient search time, theoretically guarantee convergence to the global optimum (Bonami et al., 2022; Laborie et al., 2018). While global convergence guarantees for general neural networks remain challenging, recent theoretical analysis by Fan et al. (2020) establishes that DRL can converge to a solution with a bounded error under the assumption of sufficient network capacity. Furthermore, by delegating the final scheme to the CP solver, the framework ensures that any generated solution strictly satisfies all operational constraints, thereby mitigating the risk of infeasibility. As demonstrated in Fig. 10, the hierarchical framework proposed in this study can reach the lower bound for small and medium-sized instances, indicating optimality. Although the computational difficulty scales with problem size, the obtained solutions remain consistently close to the lower bound, verifying the near-optimality and practical efficiency of the proposed method.

From the perspective of computational complexity, since the solvers combine branch-based search with cutting planes, presolve, and heuristics or constraint propagation techniques, it is difficult to express the overall time complexity in a closed form. Therefore, approximate complexity estimations are provided below.

(a) Complexity of solving the MCNF model

The planning phase is formulated as an MIP problem. The complexity of solving it using CPLEX depends on its LP relaxation and the size of the branch search tree. The computational complexity can be approximated as $O(N_{MCNF} \times C_{LP})$, where N_{MCNF} is the number of nodes in the branch-and-bound tree (exponential in the worst case), and C_{LP} represents the computational cost of solving the LP relaxation at each node. For MCNF problems, the number of decision variables scales with $K \times E$ (where K is the number of cargo types, and E is the number of network arcs). While standard LP algorithms (e.g., Simplex or Barrier methods) have polynomial-time average-case complexity, solving large-scale MCNF instances remains computationally intensive. However, advanced techniques in CPLEX, such as presolve, cutting planes, and heuristic branching, substantially prune the search space, ensuring efficient convergence in practice despite the theoretical worst-case bounds.

(b) Complexity of solving the CP model

The scheduling phase is expressed as a CP model containing several global constraints and solved using a branch-and-propagate framework. The solver alternates between constraint propagation and search, and its computational complexity can be approximated as $O(N_{CP} \times C_{Prop})$, where N_{CP} is the number of search tree nodes, and C_{Prop} represents the complexity of constraint propagation at each node. Although the theoretical worst-case complexity increases exponentially with the number of interval variables, constraint propagation and pruning rules can substantially reduce the effective search space. According to Laborie et al. (2018), propagation algorithms for interval variables typically scale between $O(n \log n)$ and $O(n^2)$, where n is the number of interval variables.

(c) Complexity of the DQN module

According to Algorithm 2, the training complexity of the DQN agent primarily depends on the complexity of the neural network, and can be expressed as $O(T_{train} \times B \times L \times N^2)$, where T_{train} is the number of training steps, B is the batch size, L is the number of layers in the neural network, and N^2 represents the matrix-vector multiplications between adjacent fully connected layers of size N. During the online decision-making phase, only one forward propagation and an action-mask computation are required, with a complexity of $O(L \times N^2 + M \times A)$, where A is the action space size, and M denotes the computational cost of the validity check for a single action.

Since DQN training is performed offline and its inference cost is negligible, the runtime complexity of the hierarchical framework is dominated by the competition between the planning (MCNF) and scheduling (CP) phases: $O(\max\{N_{MCNF} \times C_{LP}, N'_{CP} \times C_{Prop}\})$. In the context of the IPSP, the MCNF model operates at a macroscopic flow level and can typically be solved efficiently by modern MIP solvers. In contrast, the CP model handles microscopic scheduling constraints where the combinatorial complexity is most acute. Consequently, the scheduling phase constitutes the primary computational bottleneck, and the overall system complexity simplifies to $O(N'_{CP} \times C_{Prop})$. Here, N'_{CP} denotes the reduced number of search nodes after the DQN agent fixes the key decision variables. By fixing these key decisions, the DQN agent effectively reduces the CP search space and shortens the overall solution time while maintaining near-optimal solution quality, as verified by the experimental results in Table 10.

6. Computational experiments

This section outlines the experimental development environment and the parameter settings derived from our port survey. It details the generation rules for the experimental dataset and validates the feasibility of the proposed hierarchical solution framework and methods by analysing the experimental results and comparing them with real port operation data. The efficiency

of the method is further verified through comparisons with other algorithms. Additionally, this paper examines the sensitivity of the model, the port management strategies, and the impact of the proposed approach on stakeholders.

6.1. Experimental settings and test instances

The implementation in this paper was carried out through a Java project that integrates a CPLEX solver and a CP solver, alongside a Python project for the DQN models that are implemented in PyTorch. The solvers used are IBM ILOG CPLEX and CP Optimizer 22.1.1 (IBM Corporation 2022, commercial products under an academic license). While the CP Optimizer is less renowned in the scheduling domain compared to other solvers, its straightforward and expressive modelling language delivers strong out-of-the-box performance for solving real-world problems (Laborie et al., 2018). The memory usage of the solvers during testing remained below 200 MB. All experiments were conducted on a Mac Pro A2338 workstation, equipped with two 3.2 GHz Quad-Core Apple M1 processors (totalling eight cores), 8 GB of unified memory, and 256 GB of internal storage. The system operated on macOS Monterey 12.3.

The DQN training process was conducted on a PC equipped with an Intel Core i9-13900k CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU. The SGD is applied as the default optimiser, with a learning rate of 0.0001. Additionally, the DQN is trained for 200 epochs with a batch size of 8. Depending on the problem size and hardware, training these DQN models takes between 3 to 5 hours. The training dataset comprises a total of 53646 instances, spanning five data scales. The training dataset is generated as follows: a set of instances is created based on the rules for generating the experimental test dataset. If the solver fails to find the optimal solution within the time limit, the current solution state is recorded as a training instance.

The experimental input parameters and simulation data generation rules, such as equipment movement speed and unit operating efficiency, were derived from our actual port survey, as summarised in Table 7. The yard layout, number of stockpiles, and equipment, along with equipment connections, were simulated to reflect the real conditions of the dry bulk port, as illustrated in Fig. 2. Due to commercial confidentiality, specific details cannot be disclosed. To minimise the impact of the warm-up phase (i.e., when yard occupancy reaches a level sufficient for operations) on the results, relevant scenarios that align with real yard operations were simulated in the dataset. This included simulating the yard inventory status to reflect actual operational conditions. Given that port managers typically receive train and vessel arrival/departure schedules at least three days in advance, the planning and scheduling horizon in this study was set to 72 hours, with minute-level precision.

The experimental data consists of two primary components. The first component includes desensitised data derived from the actual operational data of a large dry bulk port in China, encompassing 12 cases used for method validation in Table 8. The second component comprises simulation data generated to reflect the real-world operations of the dry bulk port, used for algorithm performance comparison in Table 10 and Table 11. This includes seven datasets of varying scales, each representing progressively busier operational scenarios at the port, with each dataset containing five instances. The data primarily encompasses variables such as train and vessel carrying/demand volume and arrival/departure times. The dataset generation rules are shown in Table 7.

Table 7

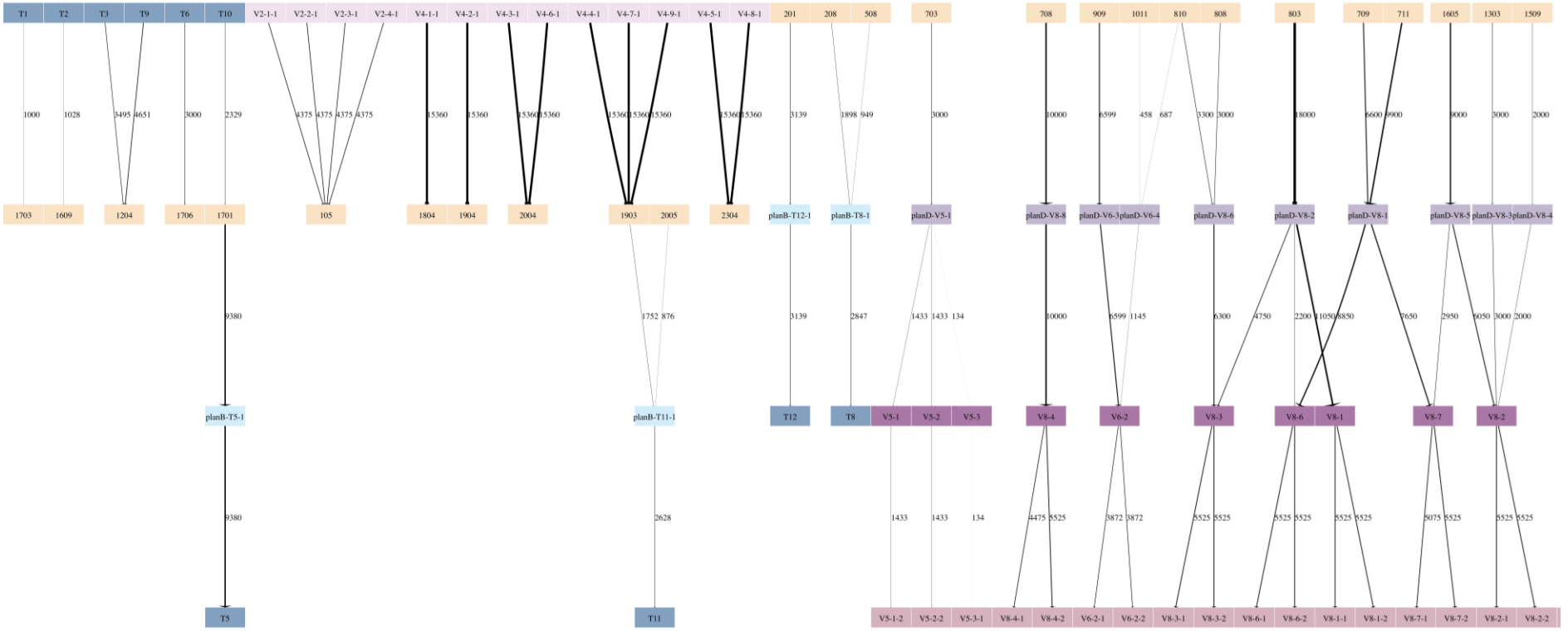
Input parameters and data generation rules.

Parameter	Value	Parameter	Value
Moving speed of stacker-reclaimers	25 m/min	Moving speed of stackers	30 m/min
Moving speed of gantry cranes	26 m/min	Moving speed of reclaimers	25 m/min
Moving speed of ship unloaders	20 m/min	Number of rails	6
Unit operation efficiency of stackers (respectively for ore, aluminium and coal)	(150, 120, 90) ton/min	Number of berths	9
Unit operation efficiency of reclaimers (respectively for ore, aluminium and coal)	(100, 80, 70) ton/min	Number of ship cabins	5–7
Unit operation efficiency of gantry cranes for ore	150 ton/min	Shipment rounds	2
Unit operation efficiency of ship unloaders for ore	50 ton/min	Inbound operation task interval	50 min
Unit operation efficiency of ship loaders for coal	100 ton/min	Outbound operation task interval	20 min
Unit operation efficiency of loading stations (respectively for ore, aluminium and coal)	(100, 70, 60) ton/min	Safety distance between devices (respectively for stackers, reclaimers, stacker-reclaimers, gantry cranes and ship unloaders)	(10, 10, 10, 2, 15) m
Unit operation efficiency of dump stations (respectively for ore, aluminium and coal)	(100, 80, 70) ton/min		
Ship loading/unloading volume distribution	[4000, 20000]:[20000, 50000]:[50000, 100000]:[100000, 200000]:[200000, 400000] = 1:2:2:3:2	Ship arrival time distribution	Normal distribution (700, 1)
Train loading/unloading volume distribution	[1000, 2000]:[2000, 3000]:[3000, 5000]:[5000, 10000] = 3:6:4:1	Train arrival time distribution	Normal distribution (70, 1)

6.2. Method validation and comparison with the manual scheduling method

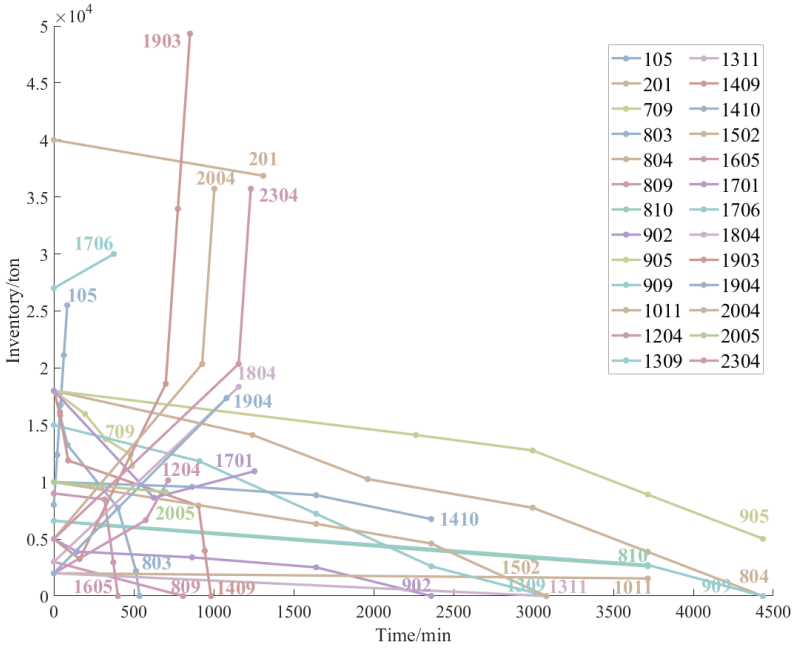
To demonstrate the effectiveness and practicality of the proposed method, this section analyses the planning and scheduling outcomes, as shown in Fig. 9. By analysing variations in stockyard inventory, equipment movement trajectories, and operation task completions, it is verified that the planning and scheduling results comply with the model constraints and fulfil the production process requirements of the multi-cargo, multi-functional dry bulk port. Fig. 9. (a) illustrates the partial allocation results of the planning phase. The diagram shows the division of train/vessel loading and unloading requirements into several operation tasks (displayed in the form of connections), the corresponding operation stockpile, and the workload associated with each task. Fig. 9. (b) depicts the majority of inventory fluctuations as the operation task progresses. The inventory remains within the range of 0 and the maximum capacity, with different stockpile types showing varying rates of increase and decrease. This variation is attributed to the differing operational efficiencies of stacking and reclaiming equipment for different cargo types. Fig. 9. (c) displays the movement trajectories of two devices on the same track, demonstrating that the devices consistently maintain a relative position and safe distance. Fig. 9. (d) presents the operation task scheduling scheme, showing that the two sub-tasks requiring mixing start and finish simultaneously to ensure uniform blending, as exemplified by sub-tasks of task V7-1-2.

To verify the feasibility of the proposed methods, the manual scheduling results of 12 practical cases are compared with the scheduling results obtained using the hierarchical approach. As shown in Table 8, the proposed algorithm consistently achieves optimal scheduling results within a short solution time (no more than 5 seconds), whereas manual scheduling typically requires over 20 minutes to generate a feasible planning and scheduling plan. The manual scheduling schemes are based on actual scheduling practices and equipment operation states observed in the 12 cases collected through the port survey. These schemes are based on plans developed by port planners and dispatchers, who typically rely on historical experience. Specifically, they tend to prioritise stockpiles that are easier to access and equipment with higher operational efficiency.

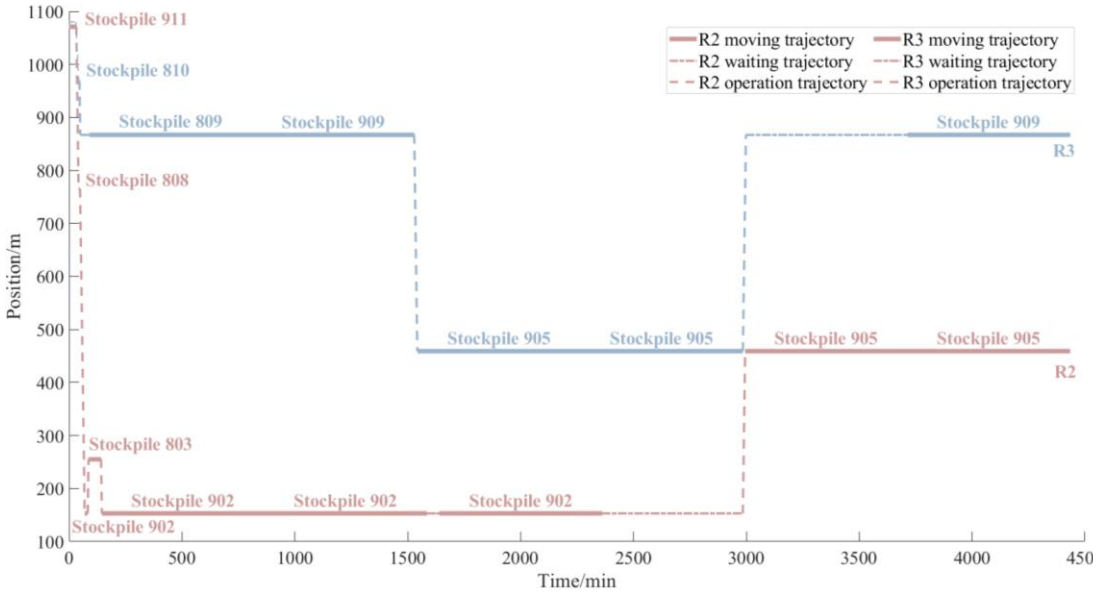


(a) Network flow allocation based on production planning and stockpile allocation

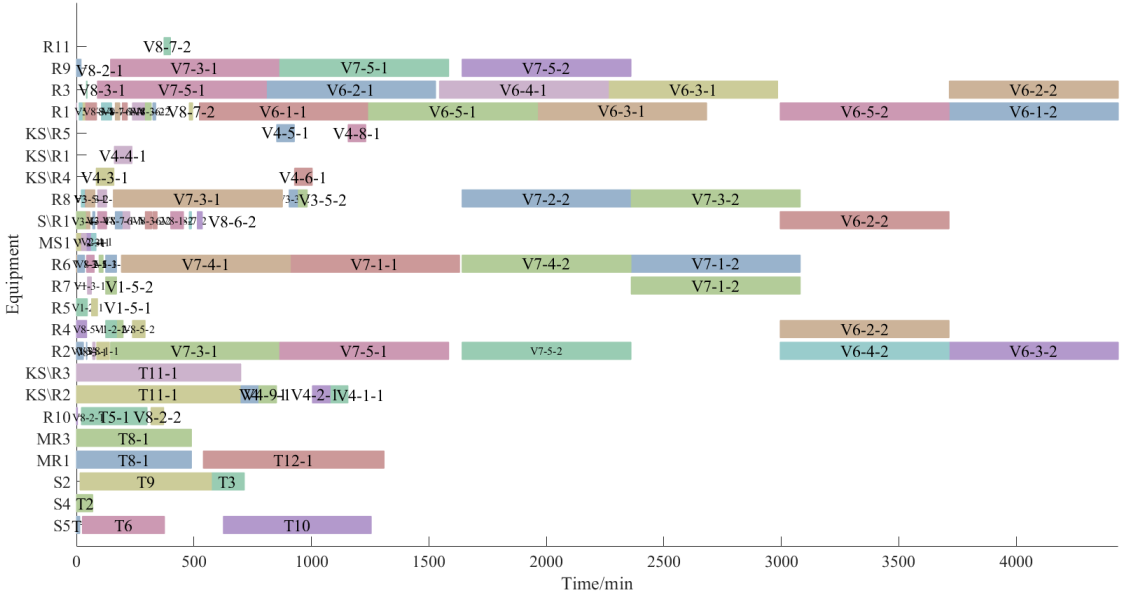
Fig. 9. Presentation of planning and scheduling results of instance 12T8V103/499.



(b) Stockpile inventory changes



(c) Equipment movement track of R2 and R3



(d) Operation task scheduling scheme

Fig. 9. (continued).

Table 8

Comparison of the manual method and the proposed method.

Case	Manual method				MCNF+CP				Relative difference (%)
	Operation time (min)	Throughput (ton)	Equipment utilisation rate (%)	Train/vessel delay rate (%)	Operation time (min)	Throughput (ton)	Equipment utilisation rate (%)	Train/vessel delay rate (%)	
R1	856	20413	15.79	50.00	51	20413	33.09	0	94.04
R2	1427	64486	25.38	50.00	623	64486	39.57	0	56.34
R3	1886	80568	16.07	50.00	865	80568	40.48	0	54.14
R4	1860	86713	19.62	25.00	1000	86713	32.35	0	46.24
R5	1944	90472	19.11	50.00	1334	90472	44.11	0	31.38
R6	2445	65426	23.67	20.00	426	65426	43.22	0	82.58
R7	559	74281	23.17	40.00	367	74281	38.63	0	34.35
R8	1844	93831	18.02	25.00	1207	93831	35.73	0	34.54
R9	2146	111396	17.04	27.27	1230	111396	36.27	0	42.68
R10	2300	107749	16.65	50.00	1144	107749	43.94	0	50.26
R11	2564	105164	17.53	18.18	1610	105164	33.44	0	37.21
R12	7866	138838	18.42	30.00	4337	138838	34.11	0	44.86

Notes: Column "Case" represents the specific port operation case analysed. Column "Operation time" reports the latest completion time for all operations within the case. Column "Throughput" represents the port's total handled cargo volume within the fixed planning and scheduling horizon, which is determined by the incoming transport demand (i.e., the loading and unloading requirements of trains and vessels). Column "Equipment utilisation rate" denotes the efficiency of equipment usage, calculated as the ratio of equipment operation time to the latest completion time. Column "Train/vessel delay rate" provides insight into the port service level, which is expressed as the ratio of the number of trains/vessels delayed leaving the port to the total number of trains/vessels arriving at the port. Column "Relative difference" indicates the comparative performance between the "MCNF+CP" method and the "Manual method," calculated as the difference between their respective latest operation completion times, normalised by the value of the former. For commercial confidentiality requirements, the data in the table is desensitised.

As shown in Table 8, the total throughput under the manual method and the one under the proposed algorithm are identical. This is because the throughput value represents the total amount of cargo handled within a fixed planning and scheduling period, which is determined by the given transport demand rather than by the scheduling strategy. The proposed algorithm does not alter the demand structure but optimises the operation sequence and equipment configuration to improve overall efficiency. Consequently, the same throughput can be completed within a shorter total operation time while achieving higher equipment utilisation. These results indicate that the manual scheduling method often leads to longer operation times and lower equipment utilisation, directly contributing to reduced port service levels—on average, 36.29% of the trains and vessels failed to depart within their planned time windows across the 12 cases examined. In contrast, the hierarchical method proposed in this study significantly improves operational efficiency and service levels, with notable improvements observed across all 12 cases.

6.3. Performance comparison with other algorithms

This section analyses the algorithmic performance from three perspectives: (1) convergence analysis of the proposed method by comparing its results with the upper and lower bounds of the integrated model obtained using CPLEX; (2) performance evaluation of the DRL module; and (3) comprehensive assessment of the proposed optimisation framework through comparative experiments with six benchmark methods.

First, to verify the convergence and near-optimality of the proposed method, this section compares the obtained solutions with the upper bounds (UBs) and lower bounds (LBs) of the integrated model solved using CPLEX, as detailed in Appendix A. As shown in Fig. 10, the results obtained by the proposed MCNF+CP+DQN method consistently fall within the bounds provided by CPLEX, achieving the optimal solution in 66.67% of the cases. As the problem size increases, CPLEX fails to compute bounds within a reasonable time limit (48 hours). Therefore, the corresponding results for medium- and large-scale instances are not presented in the figure.

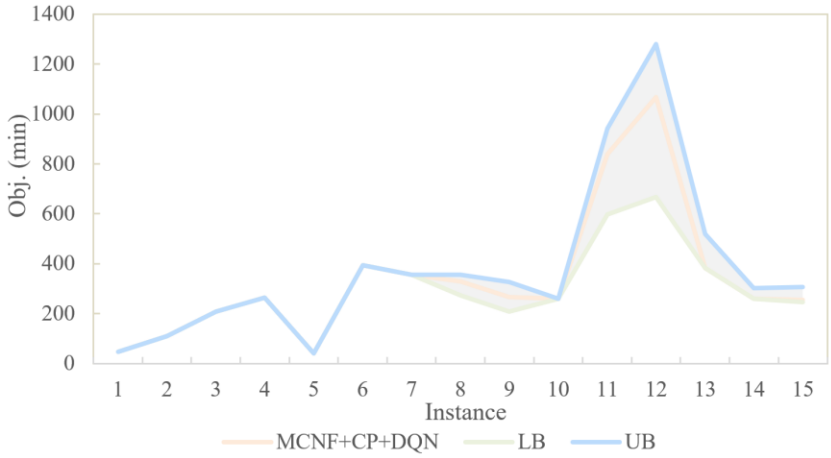


Fig. 10. Gap between CPLEX bounds and MCNF+CP+DQN's result.

Secondly, to evaluate the performance of the proposed DQN training algorithm, this study compares the reward evolution during training with two DRL algorithms, Proximal Policy Optimisation (PPO) (Jin et al., 2024) and Asynchronous Advantage Actor Critic (A3C) (Chen et al., 2025), as shown in Fig. 11. Fig. 11 presents the training trajectories of PPO, A3C, and the proposed DQN under identical conditions. All three algorithms improve their reward values over time, although their convergence behaviours differ markedly. The DQN agent exhibits the fastest reward increase, reaching a stable positive region within the first few hundred steps. PPO converges more slowly but maintains a relatively smooth learning curve, while A3C shows the slowest improvement and the largest variance. In the later training stages, DQN attains the highest average reward and the smallest oscillation range, indicating stronger convergence stability and better final policy quality. These results suggest that, for the high-dimensional discrete decision space of the IPSP, the Q-learning-based DQN is more effective than policy-gradient approaches such as PPO and A3C for learning efficient operation stream decisions. This confirms the suitability of the proposed DQN design within the DQN-CP dual-engine framework.

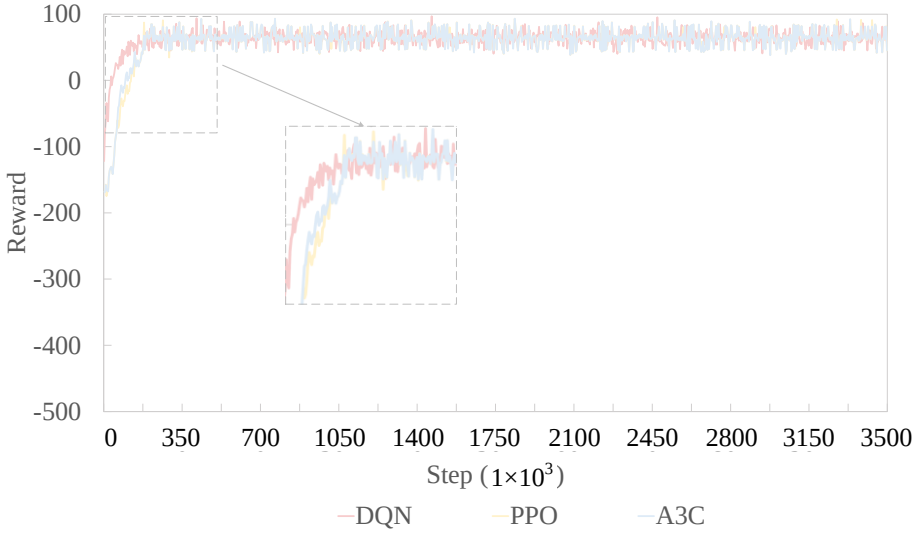


Fig. 11. Training processes of DQN, PPO and A3C.

Third, the performance of the proposed optimisation method is evaluated by comparing it with six benchmark methods: the VNS algorithm (Hansen et al., 2006), the SA algorithm (Steinbrunn et al., 1997), the PPO-SA algorithm, the A3C-heuristic (HR) algorithm, the DQN-HR algorithm and the hierarchical approach based on MCNF and CP. As classical scheduling algorithms, the first two methods have been widely applied to port scheduling (Wang et al., 2024; Thiruvady et al., 2024; Hao

et al., 2023), demonstrating notable effectiveness and efficiency in solving port management problems. However, due to the complexity of IPSP, heuristic methods such as genetic algorithm (GA) and VNS often require intricate and time-consuming repair procedures to handle infeasible solutions, making it difficult to obtain feasible results within a reasonable time limit and hindering direct comparison (Andrade et al., 2017; Lu et al., 2024). To overcome this, the MCNF+CP solution framework is employed to generate an initial feasible solution, which is then refined using VNS and SA search frameworks, serving as comparison algorithms in this study. Similarly, applying PPO and A3C directly to port operation scheduling requires complex feasibility checks for actions, leading to pronounced oscillations and non-convergence during training (Jin et al., 2024; Wang et al., 2024a). Following Jin et al. (2024) and Yang et al. (2025), this study therefore designs three hybrid variants for comparison: a PPO-SA algorithm, an A3C-HR algorithm, and a DQN-HR algorithm. Furthermore, the CPLEX solver is employed to solve the relaxed MIP model presented in Appendix A to obtain the lower bound of the problem. Because the full MIP model tends to suffer from memory overflow and infeasibility issues, the equipment, stockpile occupancy, and sequencing constraints are relaxed to derive a computationally tractable lower bound (with solution times ranging from 20 minutes to two days). It is worth noting that this relaxation yields a lower bound smaller than the true optimal solution. The detailed algorithm designs and parameter settings follow the literature (Chen et al., 2025; Hao et al., 2023; Wang et al., 2024a; Xu et al., 2020) and are summarised in Table 9.

Table 9

Parameter settings for compared algorithms.

Algorithms	Parameter
VNS	$Iter = 10$
SA	$T_{start} = 2000, T_{end} = 0.001, \mu = 0.95, L = 100$
PPO	$Iter = 3000, LR = 0.0003, \gamma = 0.99, cs = 0.2, vf = 0.5, gn = 0.5, bs = 64$
A3C	$Iter = 2800, LR = 0.00005, \gamma = 0.99, vf = 0.5$

Notes: $Iter$ is the maximum number of iterations, T_{start} and T_{end} denote the initial and ultimate temperatures, μ indicates the cooling rate, L signifies the number of iterations at each temperature, LR denotes the learning rate, γ is the discount factor, cs is the PPO clipped surrogate ratio, vf denotes the value function coefficient, gn denotes the maximum gradient norm, and bs signifies the batch size.

To account for the inherent randomness of heuristic algorithms, each algorithm was tested 10 times under identical settings. For instances where the optimal solution could not be obtained within a reasonable time, the solution time was limited to 60 minutes. Table 10 reports the average objective values, average computation times, and average gap values of the two heuristic algorithms and the two hierarchical methods proposed in this study, with the best results for each combination highlighted in bold. Table 11 presents the corresponding statistics for the three hybrid DRL methods. Several notable observations can be drawn from the results.

(1) Performance analysis of the hierarchical approach based on MCNF and CP

The proposed MCNF+CP hierarchical approach demonstrates strong performance in solving small-scale instances. For example, in the 1T1V instance, the MCNF+CP method achieves the optimal solutions within 5 seconds. However, as the problem scale increases, its solution advantage declines, with the average solution gap increasing from 0 to 72.67%. Nonetheless, 31.43% of the instances attain the optimal solution, highlighting the universality and practicality of the proposed framework. Furthermore, the solutions obtained by the relaxed lower bound (LB) and the hierarchical approach are closely aligned in most cases. For small-scale cases, the hierarchical solution matches the optimal solution obtained by CPLEX. This consistency indicates that the hierarchical method effectively reduces the search space while preserving optimality, validating its ability to balance solution quality and computational efficiency.

Table 10

Results of comparison with heuristic algorithms.

Index	Instance	LB	MCNF+CP			MCNF+CP+VNS			MCNF+CP+SA			MCNF+CP+DQN		
		Obj. (min)	Obj. (min)	Time (s)	Gap (%)	Obj. (min)	Time (s)	Gap (%)	Obj. (min)	Time (s)	Gap (%)	Obj. (min)	Time (s)	Gap (%)
1	1T1V7/28	27	46	1.89	0	46	2.27	0	46	2.88	0	**46**	**1.31**	**0**
2	1T1V10/30	99	109	0.62	0	109	1.48	0	109	1.64	0	**109**	**0.24**	**0**
3	1T1V20/44	162	209	0.76	0	209	1.40	0	209	3.73	0	**209**	**0.49**	**0**
4	1T1V17/37	123	265	1.23	0	265	1.64	0	265	3.81	0	**265**	**0.96**	**0**
5	1T1V10/24	29	40	0.15	0	40	1.09	0	40	1.54	0	**40**	**0.02**	**0**
6	3T2V31/70	384	394	62.54	0	394	47.89	0	394	56.97	0	**394**	**37.96**	**0**
7	3T2V37/88	345	355	15.78	0	**355**	**9.77**	**0**	355	12.71	0	355	10.23	0
8	3T2V25/334	239	**329**	-	**20.97**	**329**	-	**20.97**	**329**	-	**20.97**	**329**	-	**20.97**
9	3T2V27/127	149	**266**	-	**47.37**	**266**	-	**47.37**	**266**	-	**47.37**	**266**	-	**47.37**
10	3T2V36/92	179	260	0.32	0	260	2.56	0	260	5.59	0	**260**	**0.09**	**0**
11	6T4V48/250	423	**838**	-	**43.91**	**838**	-	**43.91**	**838**	-	**43.91**	**838**	-	**43.91**
12	6T4V59/204	895	**1067**	-	**67.20**	**1067**	-	**67.20**	**1067**	-	**67.20**	**1067**	-	**67.20**
13	6T4V72/365	382	382	19.53	0	382	13.73	0	382	14.92	0	**382**	**6.11**	**0**
14	6T4V32/90	178	260	0.31	0	260	3.34	0	260	4.62	0	**260**	**0.06**	**0**
15	6T4V27/58	246	256	31.36	0	256	17.29	0	256	20.59	0	**256**	**15.73**	**0**
16	9T6V85/471	2544	**3040**	-	**69.08**	**3040**	-	**69.08**	**3040**	-	**69.08**	**3040**	-	**69.08**
17	9T6V88/240	2881	3108	-	42.31	3108	-	28.22	3108	-	5.05	**3108**	**277.06**	**0**
18	9T6V87/211	3065	3322	-	25.11	3322	-	3.43	3322	-	1.96	**3322**	**376.96**	**0**
19	9T6V107/338	3704	4578	-	68.73	4578	-	68.37	4578	-	68.37	**4456**	-	**63.87**
20	9T6V78/196	4320	4813	-	66.63	4813	-	66.63	4813	-	66.63	**4704**	-	**46.70**
21	12T8V86/228	4636	4716	-	65.82	4716	-	65.82	4716	-	65.82	**4696**	-	**65.08**
22	12T8V103/499	3623	4564	-	68.60	4445	-	64.20	4445	-	64.20	**4435**	-	**63.83**
23	12T8V83/750	4103	4504	-	51.15	4216	-	47.82	4504	-	51.15	**4191**	-	**47.51**
24	12T8V103/769	5258	9126	-	51.97	9018	-	50.17	9072	-	51.07	**8969**	-	**49.36**
25	12T8V90/208	6598	7433	-	58.99	7433	-	58.99	7433	-	58.99	**7418**	-	**58.67**
26	20T10V131/432	4676	**5965**	-	**71.67**	**5965**	-	**71.67**	**5965**	-	**71.67**	**5965**	-	**71.67**
27	20T10V169/401	3952	5698	-	56.87	5660	-	55.95	5698	-	56.87	**5538**	-	**53.27**
28	20T10V158/472	5796	7332	-	56.57	7312	-	56.46	7320	-	56.5	**7180**	-	**55.65**
29	20T10V126/440	5752	7796	-	61.42	7608	-	60.46	7796	-	61.42	**7592**	-	**56.80**
30	20T10V151/446	5469	7958	-	71.54	7958	-	71.54	7958	-	71.54	**7950**	-	**71.52**
31	30T20V236/1050	6287	**7755**	-	**73.40**	**7755**	-	**73.40**	**7755**	-	**73.40**	**7755**	-	**73.40**
32	30T20V205/881	7529	9594	-	74.09	9594	-	74.09	9594	-	74.09	**9590**	-	**74.06**
33	30T20V251/857	8619	**11709**	-	**76.17**	**11709**	-	**76.17**	**11709**	-	**76.17**	**11709**	-	**76.17**
34	30T20V260/1181	9615	11759	-	76.72	11759	-	76.72	11759	-	76.72	**11309**	-	**75.72**
35	30T20V236/1035	9811	12990	-	62.96	12981	-	62.89	12990	-	62.96	**12957**	-	**62.87**

Notes: Column "Index" denotes the identification number of each instance. Column "Instance" represents the instance used, with the naming convention indicating the number of trains, vessels, operation tasks and optional operation streams. For instance, "1T1V7/28" denotes an instance comprising one train, one vessel, seven operation tasks and 28 operation streams. Column "Obj." reports the obtained objective value, which is the latest completion time (in minutes) of all operation tasks. Column "Time" presents the computation time (in seconds) of the algorithm. A symbol "-" indicates that the algorithm failed to converge to the global optimum within the computation time limit of 3600 seconds. Column "Gap" is the optimality gap reported by the CP solver.

Table 11

Results of comparison with DRL algorithms.

Index	Instance	PPO-SA			A3C-HR			DQN-HR		
		Obj. (min)	Time (s)	Gap (%)	Obj. (min)	Time (s)	Gap (%)	Obj. (min)	Time (s)	Gap (%)
1	1T1V7/28	68	0.39	47.83	68	0.16	47.83	68	0.16	47.83
2	1T1V10/30	235	0.34	115.60	235	0.22	115.60	235	0.21	115.6
3	1T1V20/44	484	0.62	131.58	485	0.39	132.06	484	0.41	131.58
4	1T1V17/37	413	0.52	55.85	414	0.34	56.23	413	0.53	55.85
5	1T1V10/24	102	0.34	155.00	123	0.22	207.50	112	0.24	180.00
6	3T2V31/70	598	0.88	51.78	598	0.58	51.78	598	0.62	51.78
7	3T2V37/88	768	1.06	116.34	768	1.04	116.34	768	1.05	116.34
8	3T2V25/334	671	0.91	103.95	671	0.57	103.95	671	0.62	103.95
9	3T2V27/127	741	0.87	178.57	741	0.83	178.57	741	0.83	178.57
10	3T2V36/92	743	1.03	185.77	743	0.65	185.77	743	0.73	185.77
11	6T4V48/250	1579	1.41	88.42	1579	0.89	88.42	1579	1.42	88.42
12	6T4V59/204	2174	1.70	103.75	2174	1.67	103.75	2174	1.11	103.75
13	6T4V72/365	1100	2.11	187.96	1100	1.74	187.96	1100	1.41	187.96
14	6T4V32/90	500	0.97	92.31	500	0.62	92.31	500	0.94	92.31
15	6T4V27/58	825	0.79	222.27	825	0.48	222.27	825	0.79	222.27
16	9T6V85/471	5454	2.48	79.41	5454	2.46	79.41	5454	1.63	79.41
17	9T6V88/240	3224	2.40	3.73	3249	1.52	4.54	3224	1.55	3.73
18	9T6V87/211	3444	2.37	3.67	3402	1.51	2.41	3644	2.37	9.69
19	9T6V107/338	5835	2.94	30.95	5116	1.79	14.81	5412	2.95	21.45
20	9T6V78/196	5481	2.14	16.52	5317	1.26	13.03	5481	2.13	16.52
21	12T8V86/228	6214	2.41	32.33	4751	1.54	1.17	4214	2.40	8.71
22	12T8V103/499	5825	2.97	31.34	5105	1.88	15.11	5105	2.84	15.11
23	12T8V83/750	9295	2.61	121.78	9281	1.55	121.45	9280	1.56	121.43
24	12T8V103/769	9940	3.15	10.83	9940	1.88	10.83	9940	1.97	10.83
25	12T8V90/208	8544	2.47	15.18	8882	1.50	19.74	8461	2.44	14.06
26	20T10V131/432	8597	3.91	44.12	7878	2.34	32.07	10037	2.63	68.26
27	20T10V169/401	7393	4.58	33.50	7394	2.87	33.51	8113	3.23	46.50
28	20T10V158/472	7978	4.24	11.11	7260	2.78	1.11	7756	3.18	8.02
29	20T10V126/440	8432	3.49	11.06	8432	2.21	11.06	8592	2.41	13.17
30	20T10V151/446	17224	6.93	116.65	17227	4.23	116.69	17224	4.70	116.65
31	30T20V236/1050	16851	6.61	117.29	16648	4.41	114.67	20242	4.51	161.02
32	30T20V205/881	15305	5.83	59.59	15621	3.55	62.89	15305	4.02	59.59
33	30T20V251/857	19489	6.83	66.44	19489	4.31	66.44	21649	4.75	84.89
34	30T20V260/1181	18364	7.41	62.38	18365	4.69	62.39	18364	7.36	62.38
35	30T20V236/1035	19040	6.63	46.95	18386	4.08	41.90	19040	4.40	46.95

Notes: Column "Gap" indicates the relative performance difference between each method and the proposed MCNF+CP+DQN method. It is calculated as the difference between their respective latest operation completion times, normalised by the completion time of the latter.

(2) Performance analysis of the MCNF+CP+VNS/SA algorithms

This paper attempted to develop solution methods based on GA and VNS separately. However, the complexity of solution construction and problem constraints prevented the random initialisation of solutions in these heuristic algorithms from generating feasible solutions within the given time. Consequently, the hierarchical approach was integrated with two heuristic algorithms, VNS and SA, to improve its performance. Experimental findings revealed that these heuristic-enhanced methods were effective in only a limited number of cases, with improvements of less than 1% and notable randomness. In certain

instances, they marginally reduce computation time; however, for cases with inherently fast convergence (e.g., 1T1V10/30), these heuristic methods actually increase computation time due to the additional operator overhead.

(3) Performance analysis of the PPO, A3C and DQN algorithms

Three comparison algorithms were developed based on DRL-heuristic hybrid designs. Although these DRL-based methods can generate rapid solutions across multiple scales, their solution quality remains poor, with average deviations of 76.49%, 78.90%, and 73.46% from the hierarchical method. This is mainly because the DRL agent incurs substantial penalty terms for infeasible actions and therefore tends to learn conservative strategies that emphasise feasibility over optimality. As a result, although DRL exhibits fast decision-making, its solution quality is unstable, with large fluctuations across instances. These findings demonstrate that using DRL alone, or combining DRL with heuristic algorithms, is insufficient to guarantee high-quality solutions for the IPSP.

(4) Performance analysis of the MCNF+CP+DQN algorithm

The proposed MCNF+CP+DQN method significantly improves performance for complex and large-scale instances, enhancing solution quality in 74.29% of the cases, with a maximum gap reduction of 42.31%. Moreover, the method markedly reduces computation time. For instance, in the 3T2V31/70 instance, MCNF+CP+DQN reduces the solution time by 64.75% relative to the baseline method, and by 26.16% and 50.08% relative to MCNF+CP+VNS and MCNF+CP+SA, respectively. Overall, these results underscore the effectiveness of the proposed method in improving both the solution quality and computational efficiency of the IPSP.

To further assess computational efficiency, Fig. 12 presents the distribution of computation time between the planning and scheduling phases when solving the IPSP using the MCNF+CP+DQN method. As illustrated, the average computation time for both phases increases with problem size. The scheduling phase generally consumes more time due to its finer-grained temporal and resource constraints, whereas the planning phase imposes a comparatively smaller computational load, accounting for an average of 3.97% of the total computation time. Nonetheless, for small- to medium-scale instances, the total computation time remains within one minute, confirming the practical feasibility and computational efficiency of the proposed hierarchical framework.

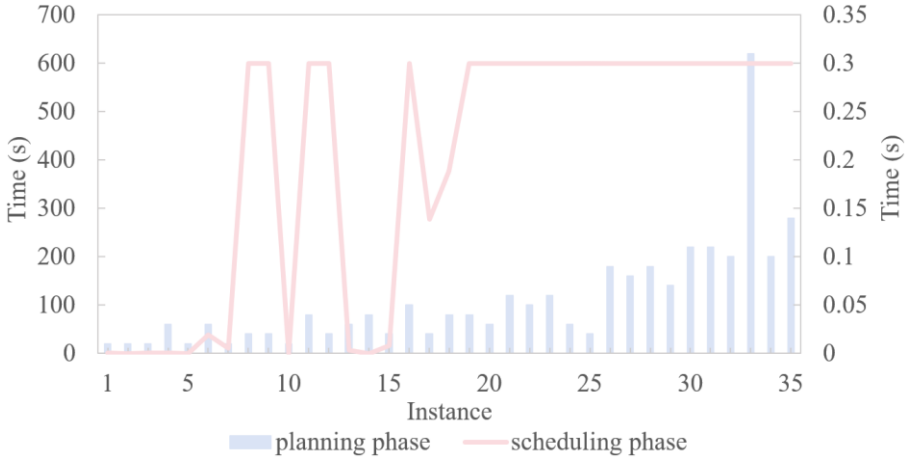
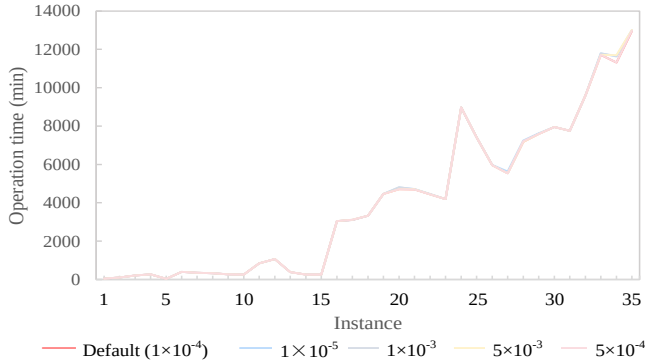


Fig. 12. Comparison of computation times in the planning and scheduling phases.

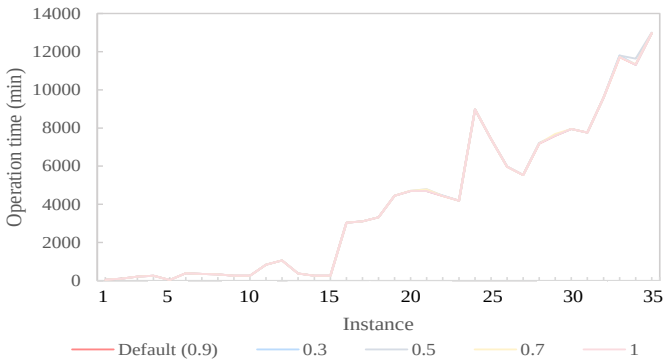
6.4. Hyperparameter and reward function experiments

To systematically evaluate the influence of hyperparameter settings and reward design on the performance of both the DQN module and the overall framework, this section conducts two groups of sensitivity experiments: (1) sensitivity analysis of hyperparameters and (2) performance analysis of the reward function.

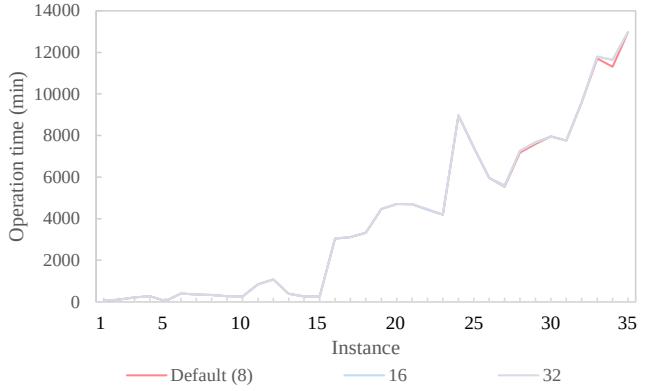
First, following Y. Zhang et al. (2024), this section examines four key hyperparameters during training: learning rate, ε-greedy exploration rate, batch size, and network structure. As shown in Fig. B. 1 of Appendix B, the learning curves exhibit rapid convergence and minimal fluctuation under different hyperparameter configurations, indicating that the DQN training process is stable and robust. This also suggests that, given the structure of the IPSP, the default reward function provides a sufficiently stable gradient signal, and the learning process is largely insensitive to fine-tuning of hyperparameters. Second, to assess the effect of hyperparameters on actual scheduling performance, this section compares the solution quality of multi-scale instances under different parameter settings (Fig. 13). The results indicate that the scheduling quality remains highly consistent across nearly all configurations, with only marginal differences observed in a few extreme cases (e.g., Instances 17 and 27). This robustness can be attributed to the dual-engine architecture proposed in this paper. On one hand, DQN is responsible solely for selecting key variables, which reduces the action space and mitigates the curse of dimensionality, leading to faster convergence. On the other hand, the downstream CP solver accurately resolves the remaining time-dependent variables, ensuring the stability of the final scheduling quality.



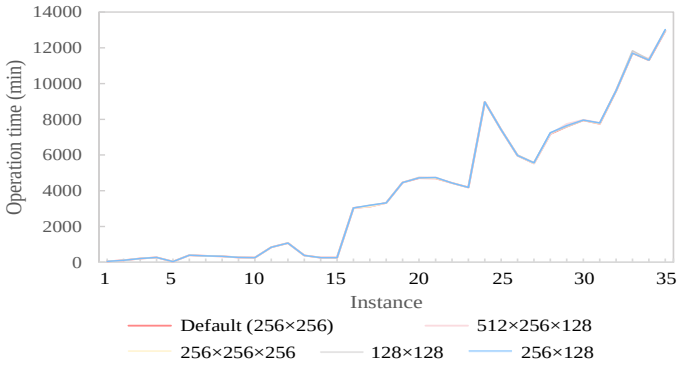
(a) Operation times under different learning rates



(b) Operation times under different ε-greedy policies



(c) Operation times under different batch sizes



(d) Operation times under different network structures

Fig. 13. Operation times under different hyperparameter settings.

The design of the reward function plays a crucial role in shaping the agent's learning behaviour. To assess the impact of reward formulation on algorithm performance, this study follows the approach of Zhang et al. (2021) and evaluates three alternative reward structures, each representing a distinct operational objective of the IPSP. Considering that port operational efficiency and service levels are reflected in vessel/train waiting time and equipment operation time, all reward functions share the same basic form as the reward function $\boldsymbol{R}^1$ (Eq. (39)): constraint violations are penalised, feasible actions are rewarded, and successful completion of all tasks yields a terminal reward to reinforce feasible full scheduling. The difference among them lies in the specific reward mechanism for task completion:

(1) $\boldsymbol{R}^2$: This reward is defined as the negative of the task completion time. This reward encourages minimising vessel/train operation time and directly reflects service-level performance.

$$\boldsymbol{R}^2(\boldsymbol{s},\boldsymbol{a}) = -100 \times \boldsymbol{et}_a \tag{44}$$

(2) $\boldsymbol{R}^3$: This reward corresponds to the vessel/train waiting time, calculated as the task start time minus the earliest start time (i.e., vessel/train arrival time). This formulation penalises start-time delays, promoting rapid task response and reducing idle waiting before execution.

$$\boldsymbol{R}^3(\boldsymbol{s},\boldsymbol{a}) = -100 \times (\boldsymbol{st}_a - est_a) \tag{45}$$

(3) $\boldsymbol{R}^4$: This reward represents equipment idle time, calculated as the difference between the start time of the current task and the end time of the previous task. This reward targets service levels and equipment utilisation efficiency, aligning with operational priorities of dry bulk ports.

$$\boldsymbol{R}^4(\boldsymbol{s},\boldsymbol{a}) = -100 \times \left(\Delta \boldsymbol{t}_{q_a^1} + \Delta \boldsymbol{t}_{q_a^2}\right) \tag{46}$$

Sensitivity analysis across 35 instances shows that all three reward function designs achieve stable convergence during training (see Fig. B. 2 in Appendix B), although their convergence speed and stability are lower than those of $\boldsymbol{R}^1$ used in this study. The solution results under different reward functions are presented in Fig. 14. The performance obtained with $\boldsymbol{R}^2$, $\boldsymbol{R}^3$, and $\boldsymbol{R}^4$ is consistently inferior to $\boldsymbol{R}^1$. The performance gap is explained by the different ways in which each reward mechanism guides policy updates. $\boldsymbol{R}^3$ leads to a short-sighted policy that encourages the agent to prioritise tasks that can be started immediately or that coincide with their arrival times. This behaviour overlooks global task criticality and often sacrifices the overall completion time in order to minimise local waiting. $\boldsymbol{R}^4$ similarly induces a locally greedy filling strategy. The agent focuses on reducing the current idle time of equipment, which tends to create resource fragmentation. As a result, critical resources become occupied by non-critical tasks and congestion is more likely to occur in later stages. The performance of $\boldsymbol{R}^2$ is closest to that of $\boldsymbol{R}^1$, although $\boldsymbol{R}^2$ functions essentially as an outcome-oriented signal that rewards decisions based purely on the completion time of the current task. By contrast, $\boldsymbol{R}^1$ incorporates a reward-shaping mechanism that provides dense and incremental feedback. This design helps guide the agent to generate smooth decision paths that respect physical and operational constraints while approaching globally optimal performance in a complex constraint space. Therefore, the results confirm the soundness and effectiveness of the reward mechanism adopted in this study.

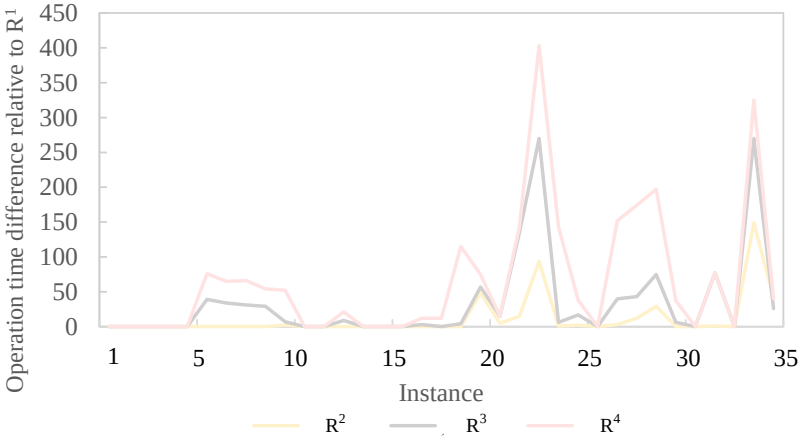


Fig. 14. Operation time differences relative to $\boldsymbol{R}^1$ under different reward functions.

6.5. Model and algorithm sensitivity analysis

Fig. 15. (a) and (b) present the comparative results of the relative changes in the objective value and the absolute changes in the solution gap with respect to the default configuration under different model parameter settings. The results indicate that relaxing several key constraints generally improves model performance, reducing the objective value and enhancing scheduling efficiency. Among these, the removal of order and non-overlap constraints has the most pronounced impact on optimisation results, with the objective value decreasing by 30–40% in some instances and even achieving optimal solutions when these

constraints are removed. This demonstrates that these two constraints play a crucial role in maintaining scheduling efficiency. In contrast, relaxing the device moving distance and crossing constraints has a relatively minor influence, suggesting that these constraints mainly affect local equipment coordination rather than overall optimisation. The results for the gap changes further show that excessive relaxation of order constraints may lead to unstable solutions or slower convergence, whereas relaxing non-overlap constraints helps obtain better solutions in certain cases. Overall, order and non-overlap constraints are key factors influencing the performance and stability of integrated scheduling, while equipment-related constraints primarily affect local operational coordination. These findings reaffirm the importance of rational constraint design in improving model effectiveness and ensuring solution robustness.

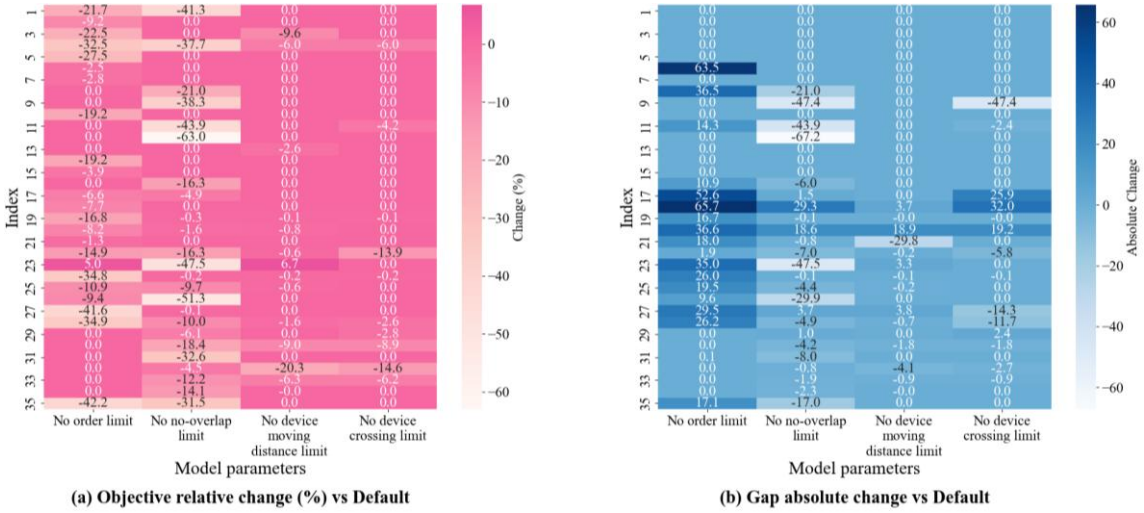


Fig. 15. Influence of model parameters on objective and Gap values.

To further evaluate the robustness of the proposed method, this section presents the results obtained under various extreme operating conditions and different instance scales. Fig. 16–Fig. 18 illustrate the changes in operation time and train/vessel delay rates under three specific scenarios: equipment capacity reduction, increased workload, and reduced train/vessel arrival intervals. It should be noted that, to ensure solvability for large-scale instances under extreme conditions, the time window constraints were relaxed during the experiments. Consequently, all instances with a non-zero delay rate represent results obtained after relaxing these constraints.

The experimental results in Fig. 16–Fig. 18 demonstrate the effects of the three key disturbance factors on port operational performance. As shown in Fig. 16, reducing equipment capacity results in a sharp increase in operation time and delay rate, particularly when capacity drops below 70% of the baseline, with an average increase in operation time of 101%. This indicates that the system is highly sensitive to equipment performance; insufficient capacity leads to task congestion and reduced scheduling efficiency. In Fig. 17, increasing workload causes a more moderate increase in operation time and delay rate, with average increases in operation time of 4.76%, 14.76%, and 21.05% as workload increases. The scheduling model maintains stable performance under moderate workload growth, highlighting the scalability and robustness of the proposed method. Fig. 18 shows that shorter arrival intervals exacerbate operational delays and extend overall operation time, especially when arrivals become highly concentrated. This underscores the importance of synchronising arrival planning with scheduling to prevent port congestion. Overall, the results indicate that equipment capacity is the dominant factor influencing system performance, while workload and arrival interval variations, though less pronounced, still have significant impacts. Ensuring sufficient equipment capacity and improving arrival coordination are therefore critical to achieving efficient and resilient port operations.

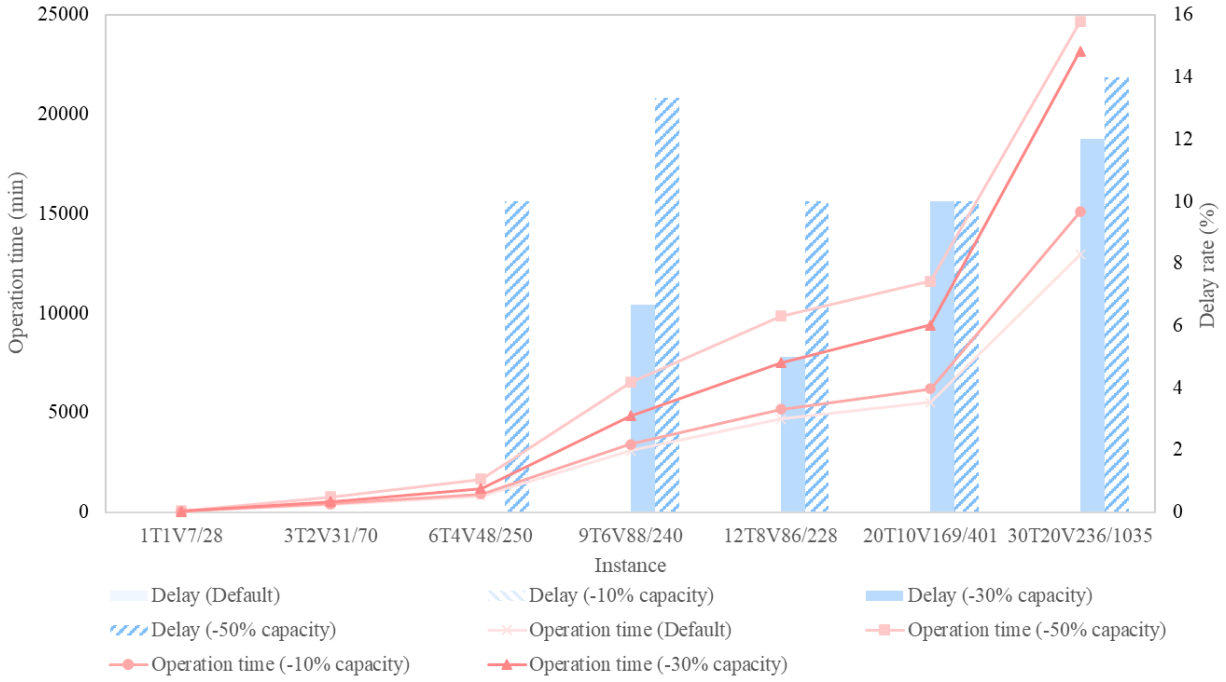


Fig. 16. Effects of equipment capacity reduction on delay rate and operation time.

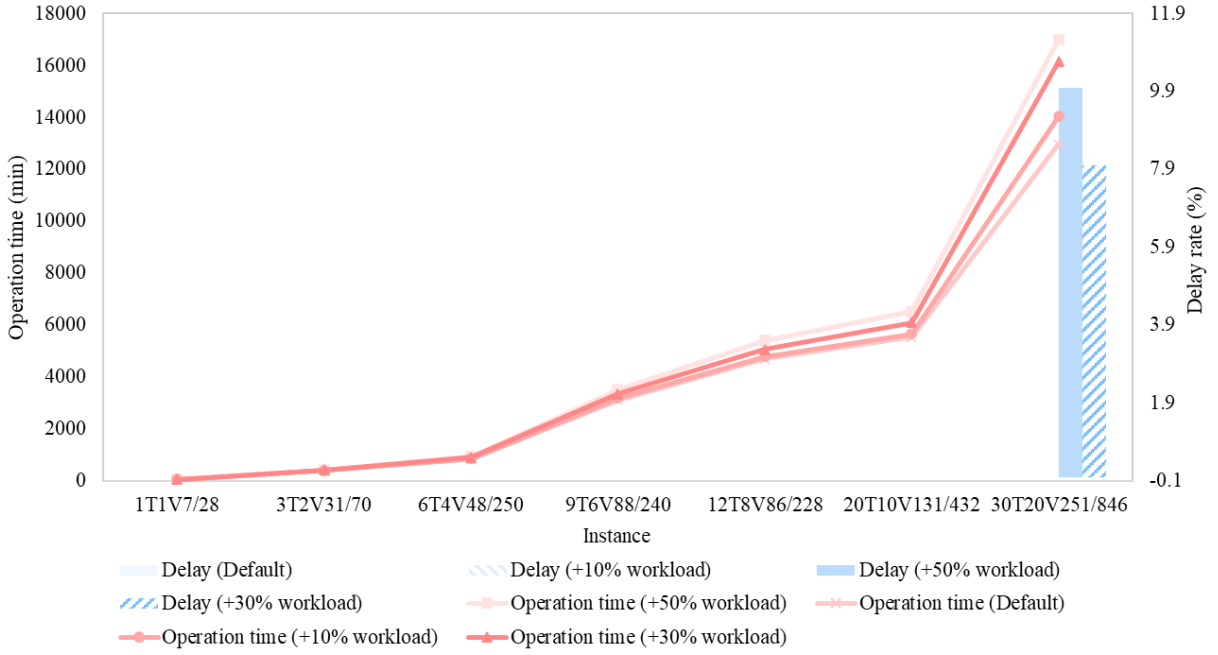


Fig. 17. Effects of increased workload on delay rate and operation time.

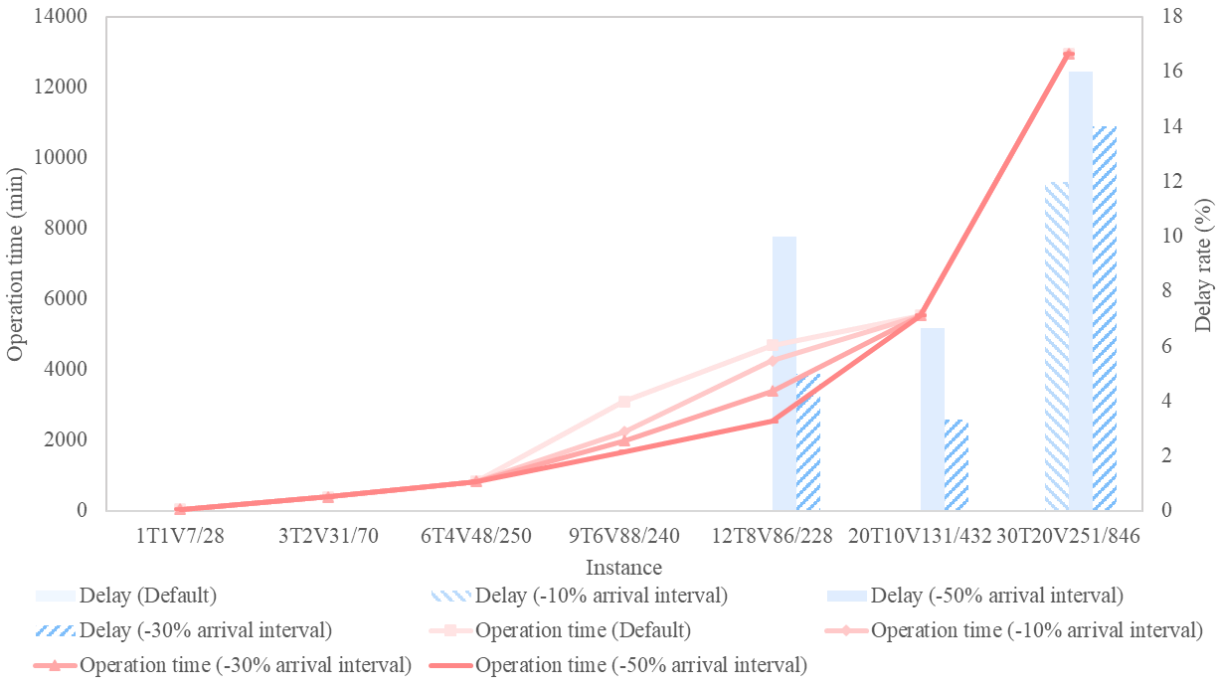
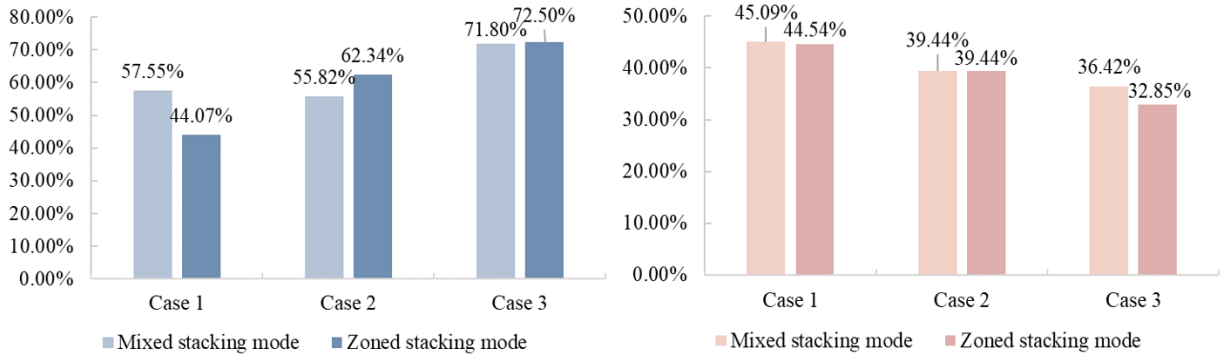


Fig. 18. Effects of reduced arrival intervals on delay rate and operation time.

6.6. Comparison of different port management strategies

This study addresses the management of multi-cargo, multi-functional dry bulk ports, where efficient yard utilisation is achieved through mixed and stacked storage of various goods. Additionally, the integration of various transportation modes can expand the service scope of the port. However, the diverse process characteristics associated with multi-type goods and multi-functional flows also present significant challenges in yard management and equipment scheduling. To provide actionable insights for dry bulk port management, this study compares multi-cargo mixed stacking and zoned stacking strategies. Using the proposed hierarchical solution method, three sets of cases with identical operation tasks but different yard management modes are solved, and the results are analysed. Given that the stacking mode has a limited impact on solution complexity and computation time, the analysis primarily focuses on equipment utilisation and stockyard utilisation under different management modes, as illustrated in Fig. 19.

The equipment utilisation rate is defined as the ratio of total equipment operation time to the latest completion time, while the yard utilisation rate is calculated as the ratio of the average inventory level to the yard's maximum inventory capacity. Fig. 19. (a) illustrates that with a smaller task scale, the zoned stacking management strategy does not yield a higher equipment utilisation rate. In such cases, the mixed stacking management mode leverages yard equipment more effectively, enhancing utilisation. However, as the task scale grows, the zoned stacking management mode demonstrates its advantage in improving equipment utilisation. For cases 2 and 3, the zoned management mode increases equipment utilisation rates by 11.68% and 0.97%, respectively. This improvement is attributed to the ability of zoned yard management to reduce equipment walking and process switching times during continuous operations. As shown in Fig. 19. (b), the zoned management mode significantly reduces yard utilisation rates, with reductions of 1.21% and 9.80% for cases 1 and 3, respectively. This indicates that zoned management mode better consolidates operations into specific stockpiles, allowing bulk cargo ports to minimise yard management area and safety inventory levels. Consequently, this approach effectively decreases the daily management workload of the yard.



(a) Comparison of equipment utilisation rate under different management modes

(b) Comparison of stockyard utilisation rate under different management modes

Fig. 19. Equipment utilisation and yard utilisation rates under different management modes.

The above findings suggest that when developing new bulk ports or upgrading existing ones, port managers should adopt management models tailored to the operational intensity and cargo types of the port. For ports with low operational volumes and fixed 1–2 cargo types, mixed stacking mode emerges as a more efficient approach. Conversely, for ports with high operational volumes and diverse cargo types, zoned stacking mode enhances equipment utilisation, minimises yard storage requirements, and improves overall management efficiency and service levels.

6.7. Implications

The research findings of this paper on production planning and scheduling for complex bulk cargo ports offer significant operational and economic benefits for multiple stakeholders:

For bulk ports, the proposed method substantially enhances operational efficiency and service levels. Compared to manual scheduling, it reduces operation time by an average of 50.72% and doubles equipment utilisation, enabling at least a twofold increase in port throughput under existing resource conditions. Furthermore, the method achieves calculation times within 60 minutes, and the training time for the DQN agent remains under 5 hours, meeting short-term scheduling requirements while effectively lowering the labour costs associated with manual scheduling.

For train companies, shipping companies, and shippers, the proposed method optimises train and vessel scheduling, significantly reducing transportation costs and enhancing operational efficiency (Zhen et al., 2022; Hu et al., 2018). Compared to manual scheduling, it achieves an average 36.29% reduction in delay rates, thereby shortening overall transportation times and improving resource utilisation.

From a broader socio-economic perspective, the proposed method strengthens port competitiveness and service levels, driving economic growth in the bulk supply chain and surrounding regions.

Moreover, the solution framework developed in this study, which addresses complex combinatorial optimisation problems such as path planning, resource allocation, and operation scheduling, demonstrates strong versatility. Its ability to handle path planning, scheduling, and optimisation makes it a valuable decision-support tool for industries requiring high-efficiency operational planning and scheduling. Table 12 provides a systematic summary of the key implications for stakeholders, highlighting the benefits and improvements achieved through the proposed method.

Table 12

Key implications for stakeholders.

Implication	Description	Stakeholders
Operational efficiency	The proposed method significantly improves port operational efficiency, reducing average operation time by 50.72% and doubling equipment utilisation, resulting in at least a twofold increase in throughput.	Port companies, port operators, shipping companies, railway companies, shippers
Cost reduction	The solution achieves computations within 60 minutes and requires less than 5 hours for agent training, eliminating the need for manual scheduling and significantly reducing labour costs. By optimising scheduling plans, the method reduces train/vessel delay rates by 36.29%, minimising operational costs and maximising resource utilisation.	Port companies, logistics companies, railway companies, shipping companies, shippers, investors
Supply chain optimisation	Enhancing port efficiency effectively resolves bottlenecks in the bulk cargo supply chain, boosting productivity and economic performance for upstream and downstream enterprises.	Bulk supply chain stakeholders, manufacturing enterprises
Economic development	Improved service levels and port competitiveness stimulate regional economic growth and drive advancements in associated industries.	Local governments, port companies, regional economic development agencies
Scalability and versatility	The proposed framework is versatile and scalable, extending its applicability to complex production scenarios such as supply chain management, warehouse management, and manufacturing processes.	Manufacturing companies, logistics service providers, warehouse operators

7. Conclusion and future work

The IPSP in large CDBPs is challenging to model cohesively and solve at large scales due to the complexity of its production processes and logistics elements. Developing an integrated model and optimisation algorithm for these ports is critical for enhancing operational efficiency and service levels in complex dry bulk environments and holds significant engineering relevance. Moreover, the modelling and solution approaches for such complex combinatorial optimisation problems offer new perspectives and methodologies for addressing similar challenges. This paper tackles these issues by constructing a hierarchical solution framework and proposing a dual-engine-driven method that combines CP with DQN, delivering an efficient solution to the integrated optimisation problem in dry bulk ports and advancing port efficiency and competitiveness.

This study centres on constructing a hierarchical solution framework and developing an optimisation algorithm tailored to the complexities of integrated production planning and scheduling in large CDBPs. The hierarchical framework decomposes a complex model that is challenging to solve as a single entity into two sub-models aligned with the production process, incorporating constraint checks to ensure consistency and solution validity across the sub-models. In terms of optimisation, an innovative dual-engine-driven approach is introduced, combining operations research optimisation with machine learning. This method significantly enhances large-scale solution efficiency by integrating CP solvers with a DQN agent. By guiding key variable decisions, the DQN helps the solver escape local optima, significantly improving solution efficiency and quality. Experimental and strategy analyses based on real-world port cases provide practical decision-making support and management insights, offering a viable path for ports to improve operational efficiency and service levels.

Potential avenues for further investigation include the following areas. First, the dual-engine-driven approach proposed in this paper exhibits certain uncertainties in improving the performance of large-scale instances. Designing more refined action spaces and interaction mechanisms to enhance the algorithm's robustness remains an area for future exploration. Secondly, the high level of uncertainty and strong interdependencies within port systems add layers of complexity, making it challenging to achieve stable solutions. Addressing uncertainties in port operations and demand could enhance the robustness of the solution approach. Furthermore, developing rescheduling strategies for ports operating under uncertain or extreme conditions represents a valuable direction for future research on complex bulk port systems.

Appendix A. Integrated MIP model for IPSP

Table A. 1 presents the notations utilised in the integrated model. Part of the symbol description has been given in Table 3 and Table 4.

Table A. 1

Notations and descriptions for the integrated model used in the IPSP.

Notation	Description
Set	
K	Set of all available operation streams, consisting of belt lines and operational equipment.
$K(i,j)$	Set of reachable operation streams between resources i and j, $i,j \in TR \cup CB(v) \cup SP$, $v \in V$.
$K(m)$	Set of all operation streams on the resource m.
$M(k)$	Set of all production resources on operation stream k.
Parameter	
d_i	Inbound or outbound volume of train/cabin batch i, $i \in TR \cup CB(v)$, $v \in V$.
est_i	The earliest operation start time of train/vessel i, which is determined by the arrival time of the train/vessel, $i \in TR \cup V$.
let_i	The latest operation end time of train/vessel i, which is determined by the departure time of the train/vessel, $i \in TR \cup V$.
ord_i	The operation sequence of train/vessel/cabin i, which is determined by the train/vessel arrival sequence or cabin loading sequence.
pos_i	Position of operation stockpile or cabin i, $i \in SP \cup C(v)$, $v \in V$.
Variable	
$\tilde{\tau}_k^{i\_sp}$	The operation end time of operation stream k that performs the inbound operation between train/cabin batch i and stockpile sp, $\tilde{\tau}_k^{i\_sp} = \tau_k^{i\_sp} + \upsilon_{i\_sp} \Big/ min\left\{ oe_m^{ca(k)} \middle| m \in M(k) \setminus \{R, SP, B\} \right\}$, $k \in K(i,sp)$, $i \in TR^+ \cup CB(v)$, $v \in V^+$, $sp \in SP$.
$\tilde{\tau}_k^{sp\_p\_i}$	The operation end time of operation stream k that performs the outbound operation between stockpile sp, blending scheme p and train/cabin batch i, $\tilde{\tau}_k^{sp\_p\_i} = \tau_k^{sp\_p\_i} + \upsilon_{sp\_p\_i} \Big/ min\left\{ oe_m^{ca(k)} \middle| m \in M(k) \setminus \{R, SP, B\} \right\}$, $k \in K(sp,i)$, $i \in TR^- \cup CB(v)$, $v \in V^-$, $sp \in SP$.
ω_{i_sp}	If inbound train/cabin batch i selects stockpile sp for the inbound operation, 1; otherwise, 0, $i \in TR^+ \cup CB(v)$, $v \in V^+$, $sp \in SP$.
ω_{p_i}	If inbound train/cabin batch i selects blending scheme p for the outbound operation, 1; otherwise, 0, $i \in TR^- \cup CB(v)$, $v \in V^-$, $p \in P$.
$\omega_k^{i\_sp}$	If operation stream k is selected to perform the inbound operation between train/cabin batch i and stockpile sp, 1; otherwise, 0, $k \in K(i,sp)$, $i \in TR^+ \cup CB(v)$, $v \in V^+$, $sp \in SP$.
$\omega_k^{sp\_p\_i}$	If operation stream k is selected to perform the outbound operation between stockpile sp, blending scheme p and train/cabin batch i, 1; otherwise, 0, $k \in K(sp,i)$, $i \in TR^- \cup CB(v)$, $v \in V^-$, $sp \in SP$.
υ_{i_sp}	The operation volume of the inbound operation between train/cabin batch i and stockpile sp, $i \in TR^+ \cup CB(v)$, $v \in V^+$, $sp \in SP$.
$\upsilon_{sp\_p\_i}$	The operation volume of the outbound operation between stockpile sp, blending scheme p and train/cabin batch i, $i \in TR^- \cup CB(v)$, $v \in V^-$, $sp \in SP$
$\tau_k^{i\_sp}$	The operation start time of operation stream k that performs the inbound operation between train/cabin batch i and stockpile sp, $k \in K(i,sp)$, $i \in TR^+ \cup CB(v)$, $v \in V^+$, $sp \in SP$
$\tau_k^{sp\_p\_i}$	The operation start time of operation stream k that performs the outbound operation between stockpile sp, blending scheme p and train/cabin batch i, $k \in K(sp,i)$, $i \in TR^- \cup CB(v)$, $v \in V^-$, $sp \in SP$
$sp(k)$	Operational stockpile of operation stream k.
inv_{sp}^{τ}	Inventory of stockpile sp at time τ.

The objective function of the integrated model is to minimise the total operation time required to complete all train and vessel operations:

$$\min \ latestEndTime \tag{A.1}$$

$$latestEndTime = \max \left(\begin{array}{l} \max \left\{ \tilde{\tau}_k^{i\_sp} \middle| i \in TR^+ \cup CB(v), v \in V^+, sp \in SP \right\}, \\ \max \left\{ \tilde{\tau}_k^{sp\_p\_i} \middle| p \in P(i), i \in TR^- \cup CB(v), v \in V^-, sp \in SP \right\} \end{array} \right) \tag{A.2}$$

The constraints of the integrated model can be categorised into nine distinct groups.

(1) Stockpile selection and loading scheme selection constraints

$$\sum_{sp \in SP} \omega_{tr\_sp} = 1 \quad \forall tr \in TR^+ \tag{A.3}$$

$$\sum_{sp\in SP}\omega_{cb\_sp}=1 \quad \forall cb\in CB(v),\forall v\in V^{+} \tag{A.4}$$

$$\sum_{p\in P(tr)}\omega_{p\_tr}=1 \quad \forall tr\in TR^{-} \tag{A.5}$$

$$\sum_{p\in P(v)}\omega_{p\_cb}=1 \quad \forall cb\in CB(v),\forall v\in V^{-} \tag{A.6}$$

(2) Operation stream selection constraints

$$\omega_{i\_sp}\geq\sum_{k\in K(i,sp)}\omega_{k}^{i\_sp} \quad \forall i\in TR^{+}\bigcup CB(v),\forall v\in V^{+},\forall sp\in SP \tag{A.7}$$

$$\omega_{p\_i}\geq\sum_{k\in K(sp,i)}\omega_{k}^{sp\_p\_i} \quad \forall sp\in SP(p),\forall p\in P(i),\forall i\in TR^{-}\bigcup CB(v),\forall v\in V^{-} \tag{A.8}$$

(3) Operation volume constraints

$$v_{i\_sp}=d_{i}\times\omega_{i\_sp} \quad \forall i\in TR^{+}\bigcup CB(v),\forall v\in V^{+},\forall sp\in SP \tag{A.9}$$

$$v_{sp\_p\_i}=\sum_{k\in K(sp,i)}d_{i}\times\omega_{k}^{sp\_p\_i}\times pro_{sp}^{p} \quad \forall sp\in SP(p),\forall p\in P(i),\forall i\in TR^{-}\bigcup CB(v),\forall v\in V^{-} \tag{A.10}$$

(4) Operation time constraints

$$\tau_{k}^{i\_sp}\geq\omega_{k}^{i\_sp}\times est_{i} \quad \forall k\in K(i,sp),\forall i\in TR^{+}\bigcup CB(v),\forall v\in V^{+},\forall sp\in SP \tag{A.11}$$

$$\tau_{k}^{sp\_p\_i}\geq\omega_{k}^{sp\_p\_i}\times est_{i} \quad \forall k\in K(sp,i),\forall sp\in SP(p),\forall p\in P(i),\forall i\in TR^{-}\bigcup CB(v),\forall v\in V^{-} \tag{A.12}$$

$$\tilde{\tau}_{k}^{i\_sp}\leq\omega_{k}^{i\_sp}\times let_{i} \quad \forall k\in K(i,sp),\forall i\in TR^{+}\bigcup CB(v),\forall v\in V^{+},\forall sp\in SP \tag{A.13}$$

$$\tilde{\tau}_{k}^{sp\_p\_i}\leq\omega_{k}^{sp\_p\_i}\times let_{i} \quad \forall k\in K(sp,i),\forall sp\in SP(p),\forall p\in P(i),\forall i\in TR^{-}\bigcup CB(v),\forall v\in V^{-} \tag{A.14}$$

(5) Operation sequence constraints

$$\tilde{\tau}_{k}^{sp\_p\_cb}\leq\tau_{k'}^{sp'\_p'\_cb'} \quad \forall k\in K(sp,cb),\forall k'\in K(sp',cb'),\forall sp\in SP(p),\forall sp'\in SP(p'),\forall p\in P(cb),\ \forall p'\in P(cb'),\left\{(cb,cb')\middle|ord_{cb'}>ord_{cb},\forall cb,cb'\in CB(v),\forall v\in V^{-}\right\} \tag{A.15}$$

$$\tilde{\tau}_{k}^{sp\_p\_i}\leq\tau_{k'}^{sp'\_p'\_i'}\vee\tilde{\tau}_{k}^{i\_sp}\leq\tau_{k'}^{i'\_sp'} \quad \forall k,k'\in K,\forall sp,sp'\in SP,\forall p,p'\in P,\left\{(i,i')\middle|ord_{i'}>ord_{i},r^{i}=r^{i'}\vee be^{i}\in be^{i'}\right\} \tag{A.16}$$

(6) Blending operation constraints

$$v_{sp\_p\_i}/v_{sp'\_p\_i}=pro_{sp}^{p}/pro_{sp'}^{p} \quad \forall k\in K(sp,i),\forall k'\in K(sp',i),\forall sp,sp'\in SP(p),\ \forall p\in P(i),\forall i\in TR^{-}\bigcup CB(v),\forall v\in V^{-} \tag{A.17}$$

$$\tau_{k}^{sp\_p\_i}=\tau_{k'}^{sp'\_p\_i} \quad \forall k\in K(sp,i),\forall k'\in K(sp',i),\forall sp,sp'\in SP(p),\forall p\in P(i),\forall i\in TR^{-}\bigcup CB(v),\forall v\in V^{-} \tag{A.18}$$

(7) Operation non-overlap constraints

$$\tilde{\tau}_{k}^{m}\leq\tau_{k'}^{m}\vee\tilde{\tau}_{k'}^{m}\leq\tau_{k}^{m} \quad \forall k,k'\in K(m),\forall m\in M \tag{A.19}$$

(8) Equipment operational constraints

$$max\left(\tilde{\tau}_{k}^{m}-\tau_{k'}^{m},\tilde{\tau}_{k'}^{m}-\tau_{k}^{m}\right)\times spd_{m}\geq\left|pos_{sp(k)}-pos_{sp(k')}\right| \quad \forall k,k'\in K(m),\forall m\in M \tag{A.20}$$

$$max\left(\tilde{\tau}_{k}^{m}-\tau_{k'}^{m'},\tilde{\tau}_{k'}^{m'}-\tau_{k}^{m}\right)\times min\left(spd_{m},spd_{m'}\right)\geq\left|pos_{sp(k)}-pos_{sp(k')}\right|+max\left(saf_{m},saf_{m'}\right) \quad \forall k\in K(m),\forall k'\in K(m'),\forall m,m'\in M(g),\forall g\in G \tag{A.21}$$

(9) Stockpile operational constraints

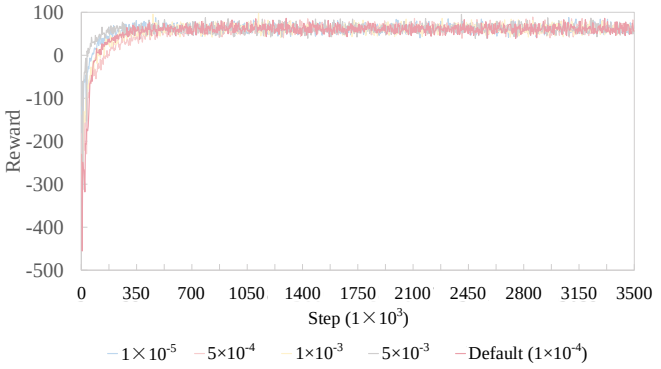
$$inv_{sp}^{\tau+1}=inv_{sp}^{\tau}+v_{sp}^{\tau}(in)-v_{sp}^{\tau}(out) \tag{A.22}$$

$$v_{sp}^{\tau}(in)=\sum_{k\in K(i,sp),\tilde{\tau}_{k}^{i\_sp}=\tau}\omega_{k}^{i\_sp}\times v_{i\_sp},\ v_{sp}^{\tau}(out)=\sum_{k\in K(sp,i),\tilde{\tau}_{k}^{sp\_p\_i}=\tau}\omega_{k}^{sp\_p\_i}\times v_{sp\_p\_i} \quad \forall sp\in SP,\forall p\in P,\forall i\in TR\bigcup CB(v),\forall v\in V,\forall\tau\in T \tag{A.23}$$

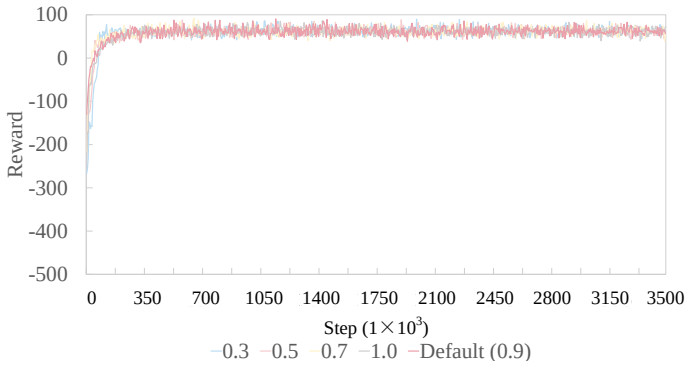
$$0\leq inv_{sp}^{\tau}\leq cap_{sp} \quad \forall sp\in SP,\forall\tau\in T \tag{A.24}$$

Appendix B. Training processes under different hyperparameter and reward function settings

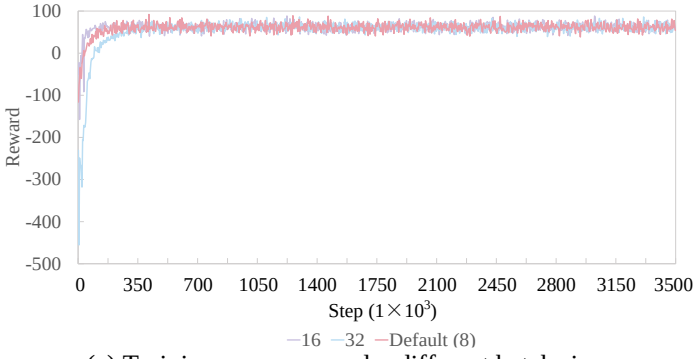
(1) Training process under different hyperparameter settings



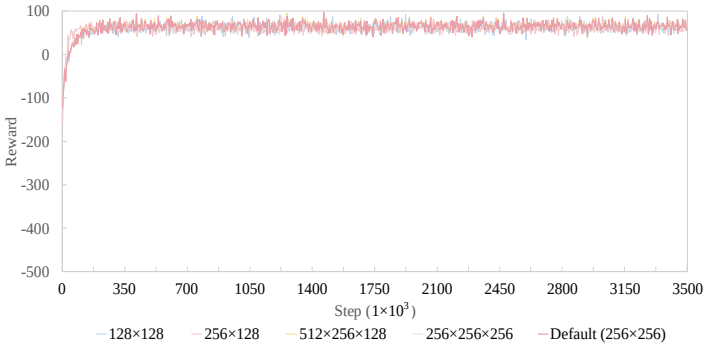
(a) Training processes under different learning rates



(b) Training processes under different ε-greedy policies



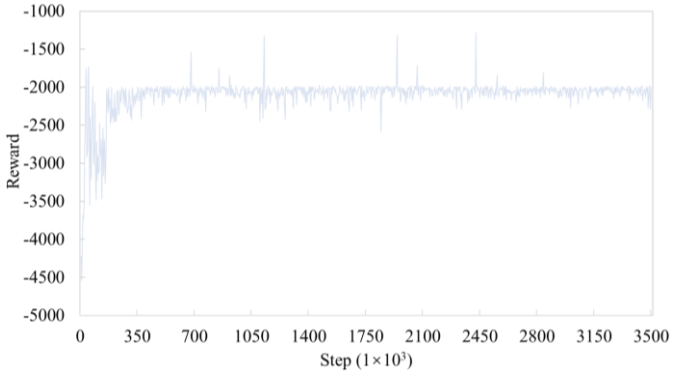
(c) Training processes under different batch sizes



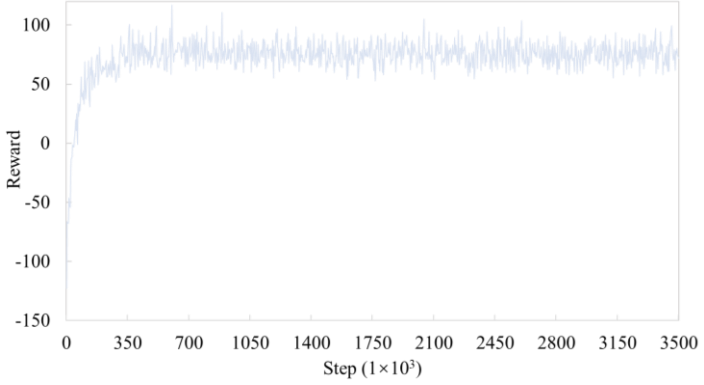
(d) Training processes under different network structures

Fig. B. 1 Training processes under different hyperparameter settings.

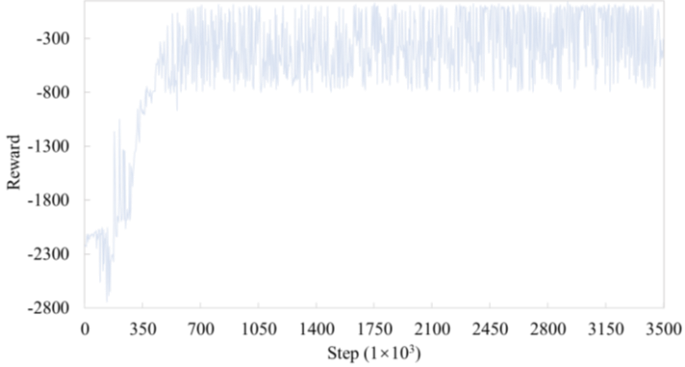
(2) Training process under different reward function settings



(a) Training process under reward function $\boldsymbol{R}^2$



(b) Training process under reward function $\boldsymbol{R}^3$



(c) Training process under reward function $\boldsymbol{R}^4$

Fig. B. 2. Training processes under different reward function settings.

References

Ai, T., Huang, L., Song, R.J., Huang, H.F., Jiao, F., Ma, W.G., 2023. An improved deep reinforcement learning approach: A case study for optimisation of berth and yard scheduling for bulk cargo terminal. Adv Eng Inform 18, 303–316. https://doi.org/10.14743/apem2023.3.474

Andrade, C.E., Ahmed, S., Nemhauser, G.L., Shao, Y., 2017. A hybrid primal heuristic for finding feasible solutions to mixed integer programs. Eur J Oper

Res 263, 62–71. https://doi.org/10.1016/j.ejor.2017.05.003
Angelelli, E., Kalinowski, T., Kapoor, R., Savelsbergh, M.W.P., 2016. A reclaimer scheduling problem arising in coal stockyard management. J Sched 19, 563–582. https://doi.org/10.1007/s10951-015-0436-y
Babu, S.A.K.I., Pratap, S., Lahoti, G., Fernandes, K.J., Tiwari, M.K., Mount, M., Xiong, Y., 2015. Minimizing delay of ships in bulk terminals by simultaneous ship scheduling, stockyard planning and train scheduling. Marit Econ Logist 17, 464–492. https://doi.org/10.1057/mel.2014.20
Belov, G., Boland, N., Savelsbergh, M.W.P., Stuckey, P.J., 2015. Exploration of models for a cargo assembly planning problem. https://doi.org/10.48550/arXiv.1504.00445
Belov, G., Boland, N., Savelsbergh, M.W.P., Stuckey, P.J., 2014. Local Search for a Cargo Assembly Planning Problem, in: Simonis, H. (Ed.), Integration of AI and OR Techniques in Constraint Programming, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 159–175. https://doi.org/10.1007/978-3-319-07046-9_12
Belov, G., Boland, N.L., Savelsbergh, M.W.P., Stuckey, P.J., 2020. Logistics optimization for a coal supply chain. J Heuristics 26, 269–300. https://doi.org/10.1007/s10732-019-09435-8
Bengio, Y., Lodi, A., Prouvost, A., 2021. Machine learning for combinatorial optimization: A methodological tour d'horizon. Eur J Oper Res 290, 405–421. https://doi.org/10.1016/j.ejor.2020.07.063
Bonami, P., Lodi, A., Zarpellon, G., 2022. A classifier to decide on the linearization of mixed-integer quadratic problems in CPLEX. Oper, Res, 70, 3303–3320. https://doi.org/10.1287/opre.2022.2267
Bouzekri, H., Alpan, G., Giard, V., 2023. Integrated laycan and berth allocation problem with ship stability and conveyor routing constraints in bulk ports. Comput Ind Eng 181, 109341. https://doi.org/10.1016/j.cie.2023.109341
Bouzekri, H., Bara, N., Alpan, G., Giard, V., 2022. An integrated Decision Support System for planning production, storage and bulk port operations in a fertilizer supply chain. Int J Prod Econ 252, 108561. https://doi.org/10.1016/j.ijpe.2022.108561
Burdett, R.L., Corry, P., Eustace, C., 2021. Stockpile scheduling with geometry constraints in dry bulk terminals. Comput Oper Res 130, 105224. https://doi.org/10.1016/j.cor.2021.105224
Burdett, R.L., Corry, P., Eustace, C., Smith, S., 2020. A flexible job shop scheduling approach with operators for coal export terminals – A mature approach. Comput Oper Res 115, 104834. https://doi.org/10.1016/j.cor.2019.104834
Burdett, R.L., Corry, P., Yarlagadda, P.K.D.V., Eustace, C., Smith, S., 2019. A flexible job shop scheduling approach with operators for coal export terminals. Comput Oper Res 104, 15–36. https://doi.org/10.1016/j.cor.2018.11.019
Cai, L., Li, W., Li, H., Zhou, B., He, L., Guo, W., Yang, Z., 2024a. Incorporation of energy-consumption optimization into multi-objective and robust port multi-equipment integrated scheduling. Transp Res Pt C-Emerg Technol 166, 104755. https://doi.org/10.1016/j.trc.2024.104755
Cai, L., Li, W., Zhou, B., Li, H., Yang, Z., 2024b. Robust multi-equipment scheduling for U-shaped container terminals concerning double-cycling mode and uncertain operation time with cascade effects. Transp Res Pt C-Emerg Technol 158, 104447. https://doi.org/10.1016/j.trc.2023.104447
Cao, Z., Sun, Q., Wang, S., 2025. Dry bulk terminal operations: A critical review of models and optimization methods. Transp Res Pt C-Emerg Technol 178, 105221. https://doi.org/10.1016/j.trc.2025.105221
Cao, Z., Wang, W., Jiang, Y., Xu, X., Xu, Y., Guo, Z., 2022. Joint berth allocation and ship loader scheduling under the rotary loading mode in coal export terminals. Transp Res Pt B-Methodol 162, 229–260. https://doi.org/10.1016/j.trb.2022.06.004
Cappart, Q., Bergman, D., Rousseau, L.-M., Prémont-Schwarz, I., Parjadis, A., 2022. Improving Variable Orderings of Approximate Decision Diagrams Using Reinforcement Learning. INFORMS J Comput 34, 2552–2570. https://doi.org/10.1287/ijoc.2022.1194
Cappart, Q., Moisan, T., Rousseau, L.-M., Prémont-Schwarz, I., Cire, A.A., 2021. Combining Reinforcement Learning and Constraint Programming for Combinatorial Optimization, in: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 3677–3687. https://doi.org/10.1609/aaai.v35i5.16484
Cheimanoff, N., Fontane, F., Kitri, M.N., Tchernev, N., 2021. A reduced VNS based approach for the dynamic continuous berth allocation problem in bulk terminals with tidal constraints. Expert Syst Appl 168, 114215. https://doi.org/10.1016/j.eswa.2020.114215
Cheimanoff, N., Fontane, F., Nour Kitri, M., Tchernev, N., 2022. Exact and heuristic methods for the berth allocation problem with multiple continuous quays in tidal bulk terminals. Expert Syst Appl 201, 117141. https://doi.org/10.1016/j.eswa.2022.117141
Chen, J., Zhu, X., Chakraborty, C., Guduri, M., Alharbi, A., Tolba, A., Yu, K., 2025. Dynamic optimization of vehicle production planning in transportation networks using federated reinforcement learning. IEEE Trans. Intell. Transp. Syst. 1–13. https://doi.org/10.1109/TITS.2024.3522523
Chen, X., He, S., Zhang, Y., Tong, L. (Carol), Shang, P., Zhou, X., 2020. Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework. Transp Res Pt C-Emerg Technol 114, 241–271. https://doi.org/10.1016/j.trc.2020.02.012
Chu, Z., Yan, R., Wang, S., 2024. Vessel turnaround time prediction: A machine learning approach. Ocean Coastal Manag 249, 107021. https://doi.org/10.1016/j.ocecoaman.2024.107021
Cimpeanu, R., Devine, M.T., O'Brien, C., 2017. A simulation model for the management and expansion of extended port terminal operations. Transp Res Pt e-Logist Transp Rev 98, 105–131. https://doi.org/10.1016/j.tre.2016.12.005
de Andrade, J.L.M., Menezes, G.C., 2023. A column generation-based heuristic to solve the integrated planning, scheduling, yard allocation and berth allocation problem in bulk ports. J Heuristics 29, 39–76. https://doi.org/10.1007/s10732-022-09506-3
de Andrade, J.L.M., Menezes, G.C., 2021. An integrated planning, scheduling, yard allocation and berth allocation problem in bulk ports: model and heuristics, in: Mes, M., Lalla-Ruiz, E., Voß, S. (Eds.), Computational Logistics. Springer International Publishing, Cham, pp. 3–20. https://doi.org/10.1007/978-3-030-87672-2_1
Derks, J.J.M., Tijs, S.H., 1985. Stable outcomes for multi-commodity flow games. Methods Oper Res 50, 493–504.
EI Mekkaoui, S., Benabbou, L., Berrado, A., 2023. Deep learning models for vessel's ETA prediction: bulk ports perspective. Flex Serv Manuf J 35, 5–28. https://doi.org/10.1007/s10696-022-09471-w
Ernst, A.T., Oğuz, C., Singh, G., Taherkhani, G., 2017. Mathematical models for the berth allocation problem in dry bulk terminals. J Sched 20, 459–473.

https://doi.org/10.1007/s10951-017-0510-8
Fan, J., Wang, Z., Xie, Y., Yang, Z., 2020. A theoretical analysis of deep Q-learning, in: Proceedings of the 2nd Conference on Learning for Dynamics and Control. Presented at the Learning for Dynamics and Control, PMLR, pp. 486–489.
Fang, Y., Wang, Y., Liu, Q., Luo, K., Liu, Z., 2021. Optimization of water resource dispatching for Huanghua Port under uncertain water usage scenario. Sci Total Environ 751, 141597. https://doi.org/10.1016/j.scitotenv.2020.141597
Ferreira, C., Figueira, G., Amorim, P., Pigatti, A., 2023. Scheduling wagons to unload in bulk cargo ports with uncertain processing times. Comput Oper Res 160, 106364. https://doi.org/10.1016/j.cor.2023.106364
Gao, Z., Ji, M., Kong, L., Hou, X., 2024. Scheduling of automated ore terminal operations based on fixed inflow rhythm. Transp Res Pt e-Logist Transp Rev 182, 103411. https://doi.org/10.1016/j.tre.2024.103411
Gao, Z., Sun, D., Zhao, R., Dong, Y., 2021. Ship-unloading scheduling optimization for a steel plant. Inf Sci 544, 214–226. https://doi.org/10.1016/j.ins.2020.07.029
Gao, Z., Zhang, M., Zhang, L., 2022. Ship-unloading scheduling optimization with differential evolution. Inf Sci 591, 88–102. https://doi.org/10.1016/j.ins.2021.12.110
Golden, B., Raghavan, S., Wasil, E. (Eds.), 2008. The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces. Springer US, Boston, MA. https://doi.org/10.1007/978-0-387-77778-8
Guo, Z., Cao, Z., Wang, W., Jiang, Y., Xu, X., Feng, P., 2021. An integrated model for vessel traffic and deballasting scheduling in coal export terminals. Transp Res Pt e-Logist Transp Rev 152, 102409. https://doi.org/10.1016/j.tre.2021.102409
Hansen, P., Mladenović, N., Urošević, D., 2006. Variable neighborhood search and local branching. Comput. Oper. Res., Part Special Issue: Constraint Programming 33, 3034–3045. https://doi.org/10.1016/j.cor.2005.02.033
Hao, L., Jin, J.G., Zhao, K., 2023. Joint scheduling of barges and tugboats for river–sea intermodal transport. Transp Res Pt e-Logist Transp Rev 173, 103097. https://doi.org/10.1016/j.tre.2023.103097
Hu, Q., Corman, F., Wiegmans, B., Lodewijks, G., 2018. A tabu search algorithm to solve the integrated planning of container on an inter-terminal network connected with a hinterland rail network. Transp Res Pt C-Emerg Technol 91, 15–36. https://doi.org/10.1016/j.trc.2018.03.019
Jiang, X., Zhong, M., Shi, J., Li, W., 2024. Optimization of integrated scheduling of restricted channels, berths, and yards in bulk cargo ports considering carbon emissions. Expert Systems with Applications 255, 124604. https://doi.org/10.1016/j.eswa.2024.124604
Jiang, Y., Cao, Z., Zhang, J., 2023. Learning to Solve 3-D Bin Packing Problem via Deep Reinforcement Learning and Constraint Programming. IEEE Trans Cybern 53, 2864–2875. https://doi.org/10.1109/TCYB.2021.3121542
Jin, J., Cui, T., Bai, R., Qu, R., 2024. Container port truck dispatching optimization using Real2Sim based deep reinforcement learning. Eur. J. Oper. Res. 315, 161–175. https://doi.org/10.1016/j.ejor.2023.11.038
Johnson, S.M., 1954. Optimal two- and three-stage production schedules with setup times included. Nav Res Logist Q 1. https://doi.org/10.1002/nav.3800010110
Koopman, B.O., 1953. The Optimum Distribution of Effort. J Oper Res Soc Am. https://doi.org/10.1287/opre.1.2.52
Krimi, I., Benmansour, R., Cadi, A.A.E., Deshayes, L., Duvivier, D., Elhachemi, N., 2019. A rolling horizon approach for the integrated multi-quays berth allocation and crane assignment problem for bulk ports. Int J Ind Eng Comput 577–591. https://doi.org/10.5267/j.ijiec.2019.4.003
Krimi, I., Todosijević, R., Benmansour, R., Ratli, M., El Cadi, A.A., Aloullal, A., 2020. Modelling and solving the multi-quays berth allocation and crane assignment problem with availability constraints. J Glob Optim 78, 349–373. https://doi.org/10.1007/s10898-020-00884-1
Laborie, P., Rogerie, J., Shaw, P., Vilím, P., 2018. IBM ILOG CP optimizer for scheduling: 20+ years of scheduling with constraints at IBM/ILOG. Constraints 23, 210–250. https://doi.org/10.1007/s10601-018-9281-x
Lazaar, N., Gotlieb, A., Lebbah, Y., 2012. A CP framework for testing CP. Constraints 17, 123–147. https://doi.org/10.1007/s10601-012-9116-0
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324. https://doi.org/10.1109/5.726791
Li, C., Wu, S., Li, Z., Zhang, Y., Zhang, L., Gomes, L., 2022. Intelligent scheduling method for bulk cargo terminal loading process based on deep reinforcement learning. Electronics 11, 1390. https://doi.org/10.3390/electronics11091390
Li, H., Jiao, H., Yang, Z., 2023. AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. Transp Res Pt e-Logist Transp Rev 175, 103152. https://doi.org/10.1016/j.tre.2023.103152
Li, H., Xing, W., Jiao, H., Yang, Z., Li, Y., 2024. Deep bi-directional information-empowered ship trajectory prediction for maritime autonomous surface ships. Transp Res Pt e-Logist Transp Rev 181, 103367. https://doi.org/10.1016/j.tre.2023.103367
Li, J., Zhang, X., Yang, B., Wang, N., 2021. Vessel traffic scheduling optimization for restricted channel in ports. Comput Ind Eng 152, 107014. https://doi.org/10.1016/j.cie.2020.107014
Li, R., Zhang, X., Jiang, L., Yang, Z., Guo, W., 2022. An adaptive heuristic algorithm based on reinforcement learning for ship scheduling optimization problem. Ocean Coastal Manag 230, 106375. https://doi.org/10.1016/j.ocecoaman.2022.106375
Li, W., Ni, S., 2022. Train timetabling with the general learning environment and multi-agent deep reinforcement learning. Transp Res Pt B-Methodol 157, 230–251. https://doi.org/10.1016/j.trb.2022.02.006
Lu, X., Zhang, Y., Zheng, L., Yang, C., Wang, J., 2024. Integrated Inbound and Outbound Scheduling for Coal Port: Constraint Programming and Adaptive Local Search. J Mar Sci Eng 12, 124. https://doi.org/10.3390/jmse12010124
Mallah, S., Aloullal, A., Kamach, O., Masmoudi, M., Kouiss, K., Chebak, A., 2023. Modeling the Bulk Port Belt-Conveyor Routing Problem Considering Interactions With Storage Spaces and Loading Operations. IEEE Access 11, 87709–87731. https://doi.org/10.1109/ACCESS.2023.3305572
Menezes, G.C., Mateus, G.R., Ravetti, M.G., 2017. A branch and price algorithm to solve the integrated production planning and scheduling in bulk ports. Eur J Oper Res 258, 926–937. https://doi.org/10.1016/j.ejor.2016.08.073

Menezes, G.C., Mateus, G.R., Ravetti, M.G., 2016. A hierarchical approach to solve a production planning and scheduling problem in bulk cargo terminal. Comput Ind Eng 97, 1–14. https://doi.org/10.1016/j.cie.2016.04.007

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529–533. https://doi.org/10.1038/nature14236

Mohajeri, M.J., de Kluijver, W., Helmons, R.L.J., van Rhee, C., Schott, D.L., 2021. A validated co-simulation of grab and moist iron ore cargo: Replicating the cohesive and stress-history dependent behaviour of bulk solids. Adv Powder Technol 32, 1157–1169. https://doi.org/10.1016/j.apt.2021.02.017

Müller, D., Müller, M.G., Kress, D., Pesch, E., 2022. An algorithm selection approach for the flexible job shop scheduling problem: Choosing constraint programming solvers through machine learning. Eur J Oper Res 302, 874–891. https://doi.org/10.1016/j.ejor.2022.01.034

Neagoe, M., Hvolby, H.-H., Taskhiri, M.S., Turner, P., 2021. Using discrete-event simulation to compare congestion management initiatives at a port terminal. Simul Modell Pract Theory 112, 102362. https://doi.org/10.1016/j.simpat.2021.102362

Pratap, S., Daultani, Y., Tiwari, M.K., Mahanty, B., 2018. Rule based optimization for a bulk handling port operations. J Intell Manuf 29, 287–311. https://doi.org/10.1007/s10845-015-1108-7

Pratap, S., Kumar B, M., Saxena, D., Tiwari, M.K., 2016. Integrated scheduling of rake and stockyard management with ship berthing: a block based evolutionary algorithm. Int J Prod Res 54, 4182–4204. https://doi.org/10.1080/00207543.2015.1111535

Pratap, S., Nayak, A., Kumar, A., Cheikhrouhou, N., Tiwari, M.K., 2017. An integrated decision support system for berth and ship unloader allocation in bulk material handling port. Comput Ind Eng 106, 386–399. https://doi.org/10.1016/j.cie.2016.12.009

Robenek, T., Umang, N., Bierlaire, M., Ropke, S., 2014. A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. Eur J Oper Res 235, 399–411. https://doi.org/10.1016/j.ejor.2013.08.015

Savelsbergh, M., Smith, O., 2015. Cargo assembly planning. EURO J Transp Logist 4, 321–354. https://doi.org/10.1007/s13676-014-0048-2

Steinbrunn, M., Moerkotte, G., Kemper, A., 1997. Heuristic and randomized optimization for the join ordering problem. The VLDB Journal 6, 191–208. https://doi.org/10.1007/s007780050040

Štepec, D., Martinčič, T., Klein, F., Vladušič, D., Costa, J.P., 2020. Machine learning based system for vessel turnaround time prediction, in: 2020 21st IEEE International Conference on Mobile Data Management (MDM). Presented at the 2020 21st IEEE International Conference on Mobile Data Management (MDM), pp. 258–263. https://doi.org/10.1109/MDM48529.2020.00060

Tang, L., Sun, D., Liu, J., 2016. Integrated storage space allocation and ship scheduling problem in bulk cargo terminals. IIE Trans 48, 428–439. https://doi.org/10.1080/0740817X.2015.1063791

Tassel, P., Gebser, M., Schekotihin, K., 2023. An End-to-End Reinforcement Learning Approach for Job-Shop Scheduling Problems Based on Constraint Programming. https://doi.org/10.48550/arXiv.2306.05747

Thiruvady, D., Nguyen, S., Sun, Y., Shiri, F., Zaidi, N., Li, X., 2024. Adaptive population-based simulated annealing for resource constrained job scheduling with uncertainty. Int J Prod Res 62, 6227–6250. https://doi.org/10.1080/00207543.2024.2311183

Tsitsiklis, J.N., Van Roy, B., 1997. An analysis of temporal-difference learning with function approximation. IEEE Trans. Autom. Control 42, 674–690. https://doi.org/10.1109/9.580874

Umang, N., Bierlaire, M., Vacca, I., 2013. Exact and heuristic methods to solve the berth allocation problem in bulk ports. Transp Res Pt e-Logist Transp Rev 54, 14–31. https://doi.org/10.1016/j.tre.2013.03.003

Wang, J., Zhang, X., Guo, W., Yang, Z., Anselem Tengecha, N., 2023. Disruption management-based coordinated scheduling for vessels and ship loaders in bulk ports. Adv Eng Inform 56, 101989. https://doi.org/10.1016/j.aei.2023.101989

Wang, W., Ding, A., Cao, Z., Peng, Y., Liu, H., Xu, X., 2024a. Deep Reinforcement Learning for Channel Traffic Scheduling in Dry Bulk Export Terminals. IEEE Transactions on Intelligent Transportation Systems 25, 17547–17561. https://doi.org/10.1109/TITS.2024.3431199

Wang, W., Liu, H., Tian, Q., Xia, Z., Liu, S., Peng, Y., 2024b. An enhanced variable neighborhood search method for refrigerated container stacking and relocation problem with duplicate priorities. Transp Res Pt e-Logist Transp Rev 188, 103643. https://doi.org/10.1016/j.tre.2024.103643

Wang, X., D'Ariano, A., Su, S., Tang, T., 2023. Cooperative train control during the power supply shortage in metro system: A multi-agent reinforcement learning approach. Transp Res Pt B-Methodol 170, 244–278. https://doi.org/10.1016/j.trb.2023.02.015

Xu, W., Yang, H., Ji, Z., Ba, M., 2024. Cognitive digital twin-enabled multi-robot collaborative manufacturing: Framework and approaches. Comput Ind Eng 194, 110418. https://doi.org/10.1016/j.cie.2024.110418

Xu, X., Wang, W., Peng, Y., Song, X., 2017. The optimaztion of train collection strategies in coal ports with eco-friendly yards: A case study in Nothern China, in: 2017 2nd International Conference Sustainable and Renewable Energy Engineering (ICSREE). Presented at the 2017 2nd International Conference Sustainable and Renewable Energy Engineering (ICSREE), IEEE, Hiroshima, Japan, pp. 58–61. https://doi.org/10.1109/ICSREE.2017.7951511

Xu, Y., Qi, L., Luan, W., Guo, X., Ma, H., 2020. Load-In-Load-Out AGV Route Planning in Automatic Container Terminal. IEEE Access 8, 157081–157088. https://doi.org/10.1109/ACCESS.2020.3019703

Yang, C., Zhang, Y., Wang, J., He, L., Wu, H., 2025. A deep reinforcement learning based multi-agent simulation optimization approach for IGV bidirectional task allocation and charging joint scheduling in automated container terminals. Comput. Oper. Res. 183. https://doi.org/10.1016/j.cor.2025.107189

Zhang, J., Li, Z., Li, L., Li, Y., Dong, H., 2021. A bi-level cooperative operation approach for AGV based automated valet parking. Transp. Res. Part C Emerging Technol. 128, 103140. https://doi.org/10.1016/j.trc.2021.103140

Zhang, X., Li, H., Sheu, J.-B., 2024. Integrated scheduling optimization of AGV and double yard cranes in automated container terminals. Transp Res Pt B-Methodol 179, 102871. https://doi.org/10.1016/j.trb.2023.102871

Zhang, X., Li, J., Yang, Z., Wang, X., 2022. Collaborative optimization for loading operation planning and vessel traffic scheduling in dry bulk ports. Adv Eng Inform 51, 101489. https://doi.org/10.1016/j.aei.2021.101489

Zhang, Y., Yang, C., Zhang, C., Tang, K., Zhou, W., Wang, J., 2024. A multi-agent reinforcement learning approach for ART adaptive control in automated

container terminals. Comput. Ind. Eng. 193, 110264. https://doi.org/10.1016/j.cie.2024.110264
Zhen, L., Zhuge, D., Wang, S., Wang, K., 2022. Integrated berth and yard space allocation under uncertainty. Transp Res Pt B-Methodol 162, 1–27. https://doi.org/10.1016/j.trb.2022.05.011